

NASA CR-139010

MAN-COMPUTER INTERACTIVE DATA ACCESS SYSTEM
(McIDAS)

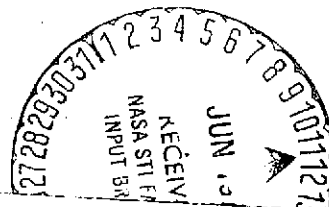
The University of Wisconsin
Space Science and Engineering Center
1225 West Dayton Street
Madison, Wisconsin 53706

18 August 1973
Final Report for Period 25 July 1972 - 24 July 1973

Reproduced by
NATIONAL TECHNICAL
INFORMATION SERVICE
U.S. Department of Commerce
Springfield, VA. 22151

Prepared for
GODDARD SPACE FLIGHT CENTER
GREENBELT, MARYLAND 20771

(NASA-CR-139010) MAN-COMPUTER INACTIVE
DATA ACCESS SYSTEM (MCIDAS) Final
Report, 25 Jul. 1972 - 24 Jul. 1973
(Wisconsin Univ.) 240 p HC \$15.00
242



N74-22856

CSCI 09B G3/08

Unclas
39273

TECHNICAL REPORT STANDARD TITLE PAGE

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle MAN-COMPUTER INACTIVE DATA ACCESS SYSTEM (McIDAS)		5. Report Date 18 August 1973	
		6. Performing Organization Code	
7. Author(s)		8. Performing Organization Report No.	
9. Performing Organization Name and Address The University of Wisconsin Space Science and Engineering Center 1225 West Dayton Street Madison, Wisconsin 53706		10. Work Unit No.	
		11. Contract or Grant No. NAS 5-21794	
12. Sponsoring Agency Name and Address Goddard Space Flight Center Greenbelt, Maryland 20771		13. Type of Report and Period Covered Final Type III 25 July 72 - 24 July 73	
		14. Sponsoring Agency Code	
15. Supplementary Notes			
16. Abstract Final report of the development of the McIDAS including brief technical and functional description of hardware and complete description of system and application software. Second section of report to include schematics, wiring lists and parts lists to be published later.			
17. Key Words (Selected by Author(s)) Data Processing. Video Displays. Satellite image processing.		18. Distribution Statement	
19. Security Classif. (of this report) U	20. Security Classif. (of this page) U	21. No. of Pages	22. Price*

PREFACE

(a) Objective

This document reports on an effort to design, develop, fabricate and test an advanced two dimensional data processing system called the Man-Computer Interactive Data Access System (McIDAS).

(b) Scope of Report

This report is a technical description of the work completed. It does not contain a narrative history of the work since that information is amply covered in monthly progress reports. The three sections of the report cover hardware, system software, and applications software. The hardware section is abbreviated and will be supplemented by complete circuit schematics, wiring lists, parts lists, component specifications, and test data to be published at a later date.

(c) Conclusions

The McIDAS equipment development required under contract NAS 5-21794 has been successfully completed.

Table of Contents

Introduction

McIDAS Hardware

Functional System Description

Theory of Operation

McIDAS System Software

Description

Program Print-outs

McIDAS Applications Software

McIDAS Cloud Tracking System

WINDCO System (Print-out)

Geosynchronous Satellite Navigation Model

NAVIGATION System (Print-out)

- The Effect of Parameterization Uncertainties on Wind Measurements

SMS Spin Axis Attitude Error Estimates

10

INTRODUCTION

At the close of the contract period all McIDAS subsystems were completed and, with some exceptions, integrated. Complete integration with software was not completed but was proceeding smoothly. Documentation of the hardware is not complete and will require about three months before all drafting work will be completed.

Rather than to wait for completion of documentation of the hardware, it was decided to publish this report and to follow with a second section at a later date. In this report is a brief overall description of the hardware with a functional block diagram and complete description of both system and application software.

-U-

McIDAS HARDWARE

Functional System Description

The McIDAS system consists of three basic sections: an access and data archive section, a control section, and a display section (See Fig. 1). The archive section consists of a video slant track recorder (modified model IVC-900) and associated control electronics which will record the SMS data to full resolution in digital format. The archive system can run 24 hours a day (except for maintenance) and can record all of the SMS data.

The control system for McIDAS is based on a Datacraft 6024/5 computer. The Datacraft is a 24-bit machine with 32,000 word storage (expandable to 128,000 words) and a basic cycle time of one microsecond. A 50 megabit digital disk, 9 track digital recorder, card reader and line printer provide conventional I/O capability. SMS Data is inputted into the Datacraft from a modified IVC-900 video slant track recorder identical to the unit used in the Archive System. This mass storage unit (see Fig. 1) can access the data on the tapes generated by the Archive System. The mass storage unit provides one full day's SMS data (visible and IR to $1/2 \times 1/2$ mile resolution) on line to the computer.

The display system accepts digital data from the Datacraft computer, converts it into standard TV type format, and presents it to a TV monitor for the operator to view. The system has the capability of "enhancing" the data as it goes to the monitor to add pseudo-color to increase the perceptual range of the operator. The system also can enhance by non-linear amplification to emphasize particular levels or areas of interest in the data.

This enhancement capability, as well as a "blending" capability which allows a number of images, each with its own enhancement, to be superimposed on the TV screen in various ways, is precisely set up and controlled by the computer working from software programs which respond to the needs of the

operator. The operator communicates with the computer via a human engineered set of controls and a language which closely resembles his own scientific background so that a minimum amount of understanding of the McIDAS implementation details is necessary. The operator then can expend his time and energy in analyzing his data instead of learning how to operate a complicated piece of electronic gear.

The basic operational approach for the McIDAS system is as follows: The operator sits before a color TV monitor. He has a keyboard and a combination of joy sticks, knobs, and switches which allow him to interact effectively with the computer. With these controls, he tells the computer the type of data, at what scale and in what format he wants to see, and the type of enhancement and blending he wants to use. The computer translates his requests into specific commands and sequences of commands which are transmitted to the display system, which in turn, presents to the color monitor the data requested in the format specified. The operator then examines the data, performs judgement, selection, or decision functions, and directs the computer to perform measurement or analyses functions on the specific data sample selected. The computer recovers the required data sample from the original data in the archive, processes the data to the precondition selected by the operator, performs the measurement or analysis function and presents the results to the operator in the format he selects. Usually the results will be in the form of another display on the color monitor.

As an example, assume the objective is to measure cloud displacement from satellite pictures. The operator would key into the computer the longitude and latitude of the area of interest, the time interval over which the measurement is to be made, and the size of the area to be displayed. The computer would then choose the appropriate satellite data, do a preliminary

navigation and display it on the TV monitor as a short repeating "movie loop" showing the dynamics of the cloud motions. The operator would then key in various color and level enhancements until he could easily see the types of clouds he was interested in--low level cumulous, for instance.

The operator would then request the computer to superimpose a cursor over the data of a particular shape, size and color. He would then position the cursor by means of a two dimensional joystick, to indicate to the computer each specific cloud he wanted analysed for displacement.

The computer would then go back to the original digital data, do a correlation on the clouds in all the pictures, calculate the displacements and do a precise navigation on the results. These results would be printed out as vectors on magnetic tape or line printer and also displayed on the TV monitor as a colored vector with proper location and orientation relative to the "loop" being displayed. The operator could then see if the results are reasonable, and if they are, take his data and leave.

Theory of Operation

The McIDAS system is made up of both purchased items and hardware which has been designed specifically for McIDAS. Purchased items are well documented by the manufacturer's publications which are included in Appendices to this report. The Datacraft computer, and its standard peripherals, are well documented and will not be detailed here. The display system, however, was designed and built almost completely at the University of Wisconsin's Space Science and Engineering Center.

The display system accepts the digital data from the computer and converts it to the standard TV format. The data are then stored on an Ampex analog video disk, an "instant replay" unit used by broadcast TV. The analog video disk records a TV frame on one circular track of the magnetic disk, then

replays this video frame back as desired at 30 frames per second. The McIDAS analog video disk has four channels, each channel has roughly 250 recording tracks, and each track can record one TV frame. For purposes of loading the analog video disk, the SMS data are segmented into 525 lines and 700 pixels per line. At each one of the locations of these pixels in the TV frame, one six-bit word is loaded from the computer in analog form. Consequently, each TV frame is broken up into a matrix of 525 by 700 or roughly 400,000 six-bit words.

For standard TV, the rate at which these pixels occur is approximately 13 megahertz. The Datacraft computer can output only 300,000 24-bit words per second. Consequently, the Video Disk cannot be loaded directly. To solve this interface problem, the Fast Storage Unit has been designed. It accepts and stores 700 six-bit words from the computer at the computer's maximum rate. Then, when the video disk head is physically over the exact location on the disk for the particular TV line this data is to comprise, it reads out the 700 six-bit words at 13 MHz, converts them to an analog video signal, and records it as a TV line. The fast storage unit then tells the computer it is ready and the computer loads another 700 words for the next TV line. In this way the entire TV frame is built up line by line. Once the TV frame is loaded, which takes about 20 to 30 seconds, the frame is then capable of being played back on the TV monitor from the analog disk.

The analog video disk has the capability of storing 250 TV frames on each of four channels, which is to say that it can store 1000 TV frames with four channels being available simultaneously. The outputs of these four channels represent the primary input to the remainder of the display system. These four inputs are routed to the signal multiplexing section, to the blending control section, and then through the level and color enhancers. The blending control section is under the control of the computer and has the capability of blending up to four channels in a number of ways: (1) By

-4-

direct super-position; i.e., two pictures can appear to be on the screen simultaneously. Each one of these pictures can have its own color enhancement and its own level enhancement. (2) By algebraically adding two frames before they are put on the color monitor. In this case, no separate level or color enhancement would be possible. (3) By matting two pictures together (i.e., a background from one picture and a foreground from another). Each one of these pictures can have its own level and color enhancement. As an example of matting, one could use the data from the infrared image to cause all of the clouds in the visible image above a certain IR brightness level to appear in one color and those below that level to appear in a different color. (4) By superpositioning grid information. This can be used, to put in overlays, outlines of geographical features, conventional weather data, outline areas of severe weather, etc. Overlay information can come in the form of another TV image, or it can come directly from the computer as digital data. In addition, a cursor, which can be in any color the operator chooses, can be positioned and its size and type changed as the operator desires. The cursor is a primary feedback method whereby the operator interacts with the computer. The operator can select data, outline areas, subtract data, change enhancements, or designate the placement of alpha-numerics all with the same cursor and appropriate directions to the computer. Note that the cursor allows the operator to do a selection not to do a measurement. The computer goes back to the original uncompromised digital data to do all measurements. Consequently, the non-linearities of the TV display do not affect quantitative data analysis.

The output of the blending control is an analog TV compatible video signal, which could go directly to the color monitor. However, in McIDAS this video signal is digitized at a 13 MHz rate, using a Computer Laboratories

HS-615 high speed analog-to-digital converter to provide the input to the enhancement section. Enhancement is done digitally because it provides much more flexibility and precision than can be achieved by analog circuits at the required frequencies and band-widths. The computer is not actively involved in the enhancement; the computer just loads the transfer function into the enhancement unit which is a look-up table of 64 six-bit word capacity. Each 6-bit digital sample of the video signal is used as an address to enter the look-up table. At the location of that address, the computer has loaded another six-bit word which represents the enhanced output value. This six-bit word is shifted out of memory, run through a high speed digital-to-analog converter and then routed out as the enhanced video signal. McIDAS includes three identical enhancement units; one look-up table is for the red color, one for green, and one for blue. Identical addresses are used for all three look-up tables and the words stored in each address contain both the color and the level enhancement information. The relative values between the three look-up tables determine the color, as seen on the monitor, and the absolute values represent the brightness.

The output of the enhancement section is three signals; red, green, and blue, which together comprise a standard RGB television video format. McIDAS uses monitors which can accept the RGB standard directly because it yields higher video resolution and bandwidth. However, a standard converter can be added which will transform the RGB format into the U.S. standard encoded broadcast TV format.

A logic interface unit between the display system and the computer, can store up to 16 24-bit computer words with provisions for expansion to 32. Each bit in these words has a specific control function in the system. The computer loads the 24-bit words into the interface along with a word address. During the vertical retrace period of the TV image the computer has time to

load over 150 24-bit words into the interface, and this is sufficient to completely change the control setup of the entire system, and also to load the look-up tables. The system responds to the operator very rapidly and can completely update the display within one vertical retrace of the TV.

The mass storage unit (see Figure 1) consists of a modified slant track recorder, and a control unit. The primary purpose of the mass storage unit is to input SMS data from the archive system to the computer. The tapes generated on the archive system's modified slant track recorder are compatible with the mass storage unit's slant track recorder. The mass storage unit can also store and retrieve data generated in the computer or from other sources (like ATS data) which feed data to the computer via special or conventional data links. The control unit converts the serial bit stream of the slant track recorder into 24-bit parallel words which can then be shuttled in and out of the computer. The control unit also accepts commands from the computer and moves the tape on the slant track recorder in accordance with instructions from the computer. The mass storage unit can be considered to be a general purpose peripheral with more than 50 billion bits of data on-line. The mass storage unit operates under the control of the computer and its specific purpose is to move data in and out of the computer.

MCIDAS SYSTEMS SOFTWARE

Objective

The McIDAS system is built around a data-craft 6024/5 computer. This computer was supplied by the manufacturer with an operating system called DMS and this operating system with only slight modifications is used to support the McIDAS software. Most of the software for this system consists of non-resident modules which are invoked by the operator or by other modules and have short lifetimes. This permits the entire operating structure to be many times larger than the computer core.

The only important modifications to the DMS monitor program itself is the addition of four functions. These functions are an interval timer multiplexor called ITM, a spooled operator output function called TQ, a transmit message to program function called SQ, and an interrogate program message function called IQ.

The interval timer multiplexor allows several programs concurrently to request interrupt routines to be entered after a given number of milliseconds.

The TQ function allows programs to pass messages to the operator without having to wait for the messages to actually occur on the operator's printer. These messages are spooled to disk and printed when their chance comes up. This implies that a task may have been completed and exited core long before the operator gets the message from that task. All messages from any McIDAS program to the operator are sent through the TQ function.

The SQ function is the basic method of inter-program communication within the system. A message containing a name of a program and a list of eight parameters is constructed and sent to the SQ function. The function receives the message and returns control immediately back to the calling program. At least 10 times per second a check is made of these messages to see if every program to which a message has been sent is presently active. If there are any programs which have had messages sent to them and which are not presently active, they are started. This allows the transmitting program, the flexibility of not having to worry about whether the receiving program is presently active. If it is not active, it will be started at a later time.

The IQ function is a method by which a program picks up traffic which was sent to it by some other program using the SQ function. The IQ function will send a program its oldest piece of traffic or an indication that there is no more traffic for that program. The normal system convention is that a program continues to pick up messages with the IQ function until it finds that there is no more traffic; at which time it exits. This is the basic non-resident nature of most of the tasks on the system.

The interval timer multiplexer is supported by a module called ITM which is permanently a resident in the monitor system. The functions TQ, SQ and IQ are partially supported by a module SYSQ which is resident in the system monitor and partially supported by a permanently active foreground routine called OPCOM. OPCOM becomes active 10 times per second, and any time a key is struck on the console typewriter. OPCOM handles spooling of operator output messages. It starts up inactive programs with messages in the SYSQ traffic pool, it examines the keys struck on the typewriter to tell whether they are important system message constructs or not, it contains the TV frame interrupt routine. Only two other significant modifications have been made in basic datacraft software. One is that the teletype handler has been modified to send individual characters struck on the teletype to OPCOM unless a legitimate system read is in progress. The other is that the magnetic tape handler has been modified so that parity errors do not cause the system to halt and query the operator. With large volumes of picture data, parity errors on tapes are inevitable and must be lived with. It is unnecessary for the operator to take positive action each time one of these occurs.

Other than these changes in the operating system monitor the rest of the McIDAS software consists of library subroutines and programs.

The following is a list of library subroutines and a brief description of their uses:

TVD is a subroutine which performs output from the computer to the TV video disk.

IK is a Fortran function which moves the bit weights within an imager to comply with the bit weight defined by portions of the McIDAS system.

9

SLEEP is a set of routines which are called by TVD and are not useful otherwise.

LTV is a function which transforms from spatial TV coordinates to time and TV coordinates. That is, it handles the interlace.

STAGE is a package which assembles and disassembles 8 bit bytes.

IO and FIO are routines which allow Fortran linkages into the input - output control system, thus opening the full resources of the monitor to Fortran programs.

WAIT is a Fortran linkage to the system wait routine. This function allows a program to suspend itself until a given event has taken place.

CSF is the name of the Fortran linkage into the functions SQ, TQ and IQ which have been defined previously.

HIO is a Fortran linkage directly into the machine level input-output command. It is used by routines to communicate with the McIDAS display system.

GETGAM is a routine which allows programs to obtain results of a navigation which have been stored in a disk file.

TVSAT is a routine which transforms from TV coordinates to spacecraft coordinates.

LOOKUP retrieves data from the frames file and the tape reels file.

LOOKUP is used by TVSAT and by other programs.

The following is a list of the nonresident programs which form the majority of the McIDAS system software:

TTY is the program which handles operator commands. TTY is call by OPCOM whenever it sees the line feed has been typed on the typewriter. TTY receives a message from the operator and encodes it into a message which is sent out through the SQ function. Therefore, through TTY the operator can send a message to any program within the McIDAS system.

10

ZLANDM FRAMES and REELS and KLUDGE are jobs which are run to initialize the three major library data files for McIDAS: the landmark file, the frames file, and the reels file. These jobs are run only once when the system is initialized. If they were run at any other time, it would cause a loss of all information which had been stored in these files.

SETKEY is a small program which allows the operators to change the state of any one of the twenty-four systems keys. Presently very few of these keys have any defined meanings.

DPROJ is a job which is run to define the legitimate projects which are allowed to use the system and to initialize the systems keys desired for that project.

DEFTAP is a routine which inserts a tape definition into the reels file.

DFRAME is a routine which inserts the definition of the contents of the TV frame into the frames file. DFRAME is normally called by the system when a frame is loaded.

WHATAP is a routine whose output is a list of the days in which at least one tape has been defined in the reel file.

DVERS is a routine which allows the user to change the version name of the McIDAS system which is typed out anytime someone begins operations.

RESFRM is a routine which restricts the projects which are allowed to write on any certain frame on the TV set. This function guarantees that users may destroy their own data but not the data of others.

LOGDUM is a program which is called once a month to dump out the contents of the usage log. This allows a permanent record of which projects have been using the system and how much time they consumed.

PRNOUT is a routine which causes the typeout of from 1 to 7 integers. This program is used both mainly for debugging.

//

ERADIR is a program which erases the directory of the 8 digital areas. This directory called DKDIR normally describes the information contained in these 8 areas.

STATUS is a program which outputs to the operator the names of all presently active programs, along with their bounds and their status.

SETF is a program which is called by the operator to specify which frame should be displayed on the TV set.

DLIM is a program through which the operator defines the bounds of a loop.

DRATE is a program through which the operator defines the rate for looping.

LOOP is a program which supports the actual looping function itself.

NAVFRM initiates a call to the navigation system to perform a navigation on the day which is presently being studied on the TV set.

DEFPNT is a mechanism through which the operator defines the point on the earth which is going to be used for navigation measurement.

DORBIT is a program which is called to define the orbital characteristics of the satellite in question on a specific day. DORBIT encodes this information and sends it to the routine DLANDM which handles all landmark, orbit, and other day-related data.

DSRATE is a routine which is used to advise the system of the spin rate of a satellite on a specific day. DSRATE also sends its data through DLANDM.

DLANDM is the routine which inserts data into the file LANDMA. Data in this file includes landmarks, spin rates, orbits, and beta values.

FINDTP is a routine which lists all tape reel definitions on any given day.

LISTAP is a routine which lists any single given tape reel definition.

12

FINFRM lists all the frame definitions which occur on any given day.

SAVEDK is a routine which causes the dist directory DKDIR and the contents of the 8 available digital data frames to be saved on a magnetic tape for future use.

RESTDK is the inverse of the SAVEDK function. It causes a previously saved set of 8 areas to be loaded back onto the digital disc. Through these two functions each user can have his own eight areas of interest with very little difficulty.

DELMRK is a routine which removes an entry from a file LANDMA. DELMRK is essentially an inverse of DLAMDM.

WCELL is a program which scans the file LANDMA to find out which cell in this file has the most entries. This allows the user to find if this file has any danger of overflowing.

EDAY is a program which eliminates all entries in file LANDMA which are associated with a given day.

LISIMK is a routine which lists data contained in the file LANDMA.

LISDIR is a routine which lists all the data in the file DKDIR. That is, it lists the contents of the 8 presently existing digital disk areas.

LFRAME is a program which lists the definition of any one particular frame in the TV system.

LNDMARK is a routine which receives a landmark measurement from the operator, encodes it into the proper format, and sends it to DLANDM for entry into the LANDMA file.

ENH001 is a program which causes a load of the enhancement tables in the McIDAS system. This allows the user to change the transfer function between the data on the disc and the data in the TV system.

LOGOUT is a program which is called by any user at the end of his use of the system. This program records that use in the log file.

LDCNTR is a routine called by the operator to perform a transfer of data from the digital tape to one of the 8 disk areas.

ABSLD is a program which is called by LDCNTR to perform the actual loading process.

LOGGIN is a program called by a user when he begins use of the system.

CNTRL is a routine which receives commands from OPCOM which have been typed in a single key strike by the operator. If the operator strikes a line feed key on the typewriter, TTY is called to read a message. If the operator strikes some other key the data from that key strike is sent to CNTRL for processing.

CC is an almost exact duplicate of the program TTY except that it receives its input from the card reader instead of the typewriter. This allows commands in the user language to be prepared ahead of time if desired.

NAMLIS is a job which is run to define the key words in the operator language. It consists of a list of cards containing two letter keys followed by the name of the program to which that key is associated. This list is saved in a file and is available to the programs TTY and CC.

LDCNTV is the program which transfers data from the digital disk to the TV system.

In addition to these programs, many applications programs are also constructed in the same way. That is, they become verbs in the operator's language and they perform their tasks as nonresident programs of short duration. Therefore, as far as the system is concerned, no distinction is made between systems programs and applications programs; all have the same structure and all have the same resources available to them.

14

\$JOB TVD
\$ASSIGN,6,0
\$ASSIGN,5,LR
\$REW,5
\$FORTRA

SUBROUTINE TVE
DIMENSION MESX(10)
DATA MFSX/6HDOIOXX/
CALL S2
CALL S0(MESX)
RETURN
END

SUBROUTINE TVI(N)
C N IS SIGNAL SYSTEM TO WRITE ON
C EVEN/ODD IS FOUND FROM FRAME POINTER
COMMON/SYSCOM/NDAT(100)
COMMON/TVL/LINE
DIMENSION MNABL(4),NABLH(2)
DATA MNABL/'00200000','00400000','01000000','02000000/
DATA NABLH/'11252500','22525200/

C MODULATOR FNABLE
CALL HIO('101','1120',MNABL(N))
C HEAD ENABLE
MF=NDAT(56)
MF=MOD(MF,2)+1
CALL HIO('101','1000+16*7,NABLH(MF))
LINE=1

C TEMPORARILY BLANK DISPLAY
CALL HIO('101','2000+16*5,0)
RETURN
END
SUBROUTINE TVD(J)
COMMON/TVL/LINE
DIMENSION J(700)
CALL S2
CALL HIO('101','1000+16*4','01000000)
ICW='1000+4*16
CALL TVX(ICW,J)
CALL HIO('101','1000+4*16','00600000+LTV(LINE))
LINE=LINE+1
CALL S1
RETURN
ENDS

\$FILEMA
ESTAB,5,TVD,0,1,0,4
EXIT
\$EOJ

15

```

$JOB IK
$ASSIGN,5,LR,6,0
$REW,5
    FUNCTION IK(J)
C MOVES BITS AROUND TO OCCUPY BINARY WEIGHTS FOR IV COLOR BRIGHTNESSES
    DIMENSION IB(18),IW(18)
    DATA IW/13,14,15,16,17,18,7,8,9,10,11,12,1,2,3,4,5,6/
    K=J
    DO 200 JA=1,18
    IB(JA)=MOD(K,2)
200  K=K/2
    IK=0
    DO 201 JA=1,18
    JB=IW(JA)
201  IK=2*IK+IB(JB)
    R=TURN
    ENDS
$FILEMA
ESTAB,5,IK,0,1,0,4
EXIT
$EOJ

```

16

\$JOB SLEEP
\$ASSIGN,6,0,5,LR
\$REW,5
\$ASSEMB

XDEF S1,S1
XDEF S2,S2
XDEF TVX,TVX

CW
TVX

GAP 1
TMA 0,1
TAM CW
GAP 1
TJM RET
TNK 699

LOOP

TMA CW
HXI
DCW '101
BNZ *-1
TMA 0,1
DOB '300
XOB '377
LLA 10
DMA ='00200000
DDW '101
BNZ *-1

LOOP2

RXI
AOI 1
BWK LOOP
TNK 323
TMA CW
HXI
DCW '101
BNZ *-1
TMA ='00200000
ODW '101
BNZ *-1

S1

RXI
NOP
BWK LOOP2
BUC* RET
TFM W
TJM RET
TLO INT
TOI 40
BLU '30

S2

BUC* RET
TJM RET
TMA W
BNN OK
TLO W
BLU \$WAI
BUC* RET

OK
RET
W
INT

DATA 0
YZM W

17

TME* DISINT
TZA
TD4
BUC 0,J
DAC '200
END\$

BISINT

\$FILEMA
ESTAB,5,SLEEP,0,1,0,4

EXIT

\$EOJ

18


```
$JOB LTV
$ASSIGN,5,LR
$REW,5
$FORTRA
    FUNCTION LTV(J)
    K=J/2
    L=MOD(J,2)
    LTV=K+L*263
    RETURN
    ENDS
$FILEMA
ESTAB,5,LTV,0,1,0,4
EXIT
$EOJ
```

\$JOB STAGE
 \$ASSIGN,6,0
 \$OPTIONS 0 7 8 23
 \$ASSIGN 5=LR
 \$REW,5
 \$ASSEMB

	IDEN	STAGE
	NAME	STAGE
	XDEF	PACK,PACK
	XDEF	CRACK,CRACK
CRACK	GAP	1
	TMA	0,I
	TAK	
	GAP	1
	TIA	
	DMA	MASK
	DMA	INST1
	TAM	PLUG1
	GAP	1
	TIA	
	DMA	MASK
	DMA	INST2
	AKA	
	NKK	
	TAM	PLUG2
	TJM	RET
	TMJ	JAD
	TOA	0
PLUG1	***	
PLUG2	***	
	BBJ	**+1
	BWK	PLUG1
	BUC*	RET
JAD	DATA	'20000000
MASK	DATA	'77777
INST1	EMB	0
INST2	TAM	0,K
INST3	TMA	0,K
INST4	RBM	0
RET	***	
*		
PACK	GAP	1
	TMA	0,I
	TAK	
	GAP	1
	TIA	
	DMA	MASK
	DMA	INST3
	AKA	
	TAM	PLUG3
	GAP	1
	TIA	
	DMA	MASK
	DMA	INST4
	NKK	
	TAM	PLUG4

20

```
TJM    RET
TMI    JAD
PLUG3  ***
PLUG4  ***
      BBI    **1
      BWK    PLUG3
      BUC*   RET
      ENDS
$FILEMA
ESTAB,5,STAGE,0,1,0,4
EXIT
$EOJ
```

21

\$JOB ITM
 \$ASSIGN,6,0
 \$OPTIONS 23
 \$ASSIGN,5,LR
 \$REW,5
 \$ASSEMB

	IDEN	INTERVAL TIMER MULTIPLEXOR
*		. ENTRY IS
*) K LOC
*		. I COUNT
*		. WILL DO BLE TO LOC IN (K)
*		. AFTER I MILLISECONDS
	XDEF	TIRTN,TIRTN
	XDEF	ITM,ITM
ISAVE	BLOK	5
TSAVE	BLOK	5
CLOK	EQIV	'77774 LOCATION OF ONE SECOND CLOCK
CCOUNT	DATA	-1000
N	DATA	0
	BLOK	16
LOC	BLOK	1
	BLOK	16
COUNT	BLOK	1
CALLER	BUC	0,K
TIRTN	***	
	TRM	ISAVE
	HIT	
	AOT	100
	RCT	
	AUM	CCOUNT
	BNZ	RACK
	AUM	CLOK
	TNA	1000
	TAM	CCOUNT
BACK	CZM	N
	BOZ	RET
	TMK	N
KLOOP	AUM	COUNT,K
	BOZ	GOT1
B	BWK	KLOOP
RET	TMR	ISAVE
	BRL*	TIRTN
GOT1	TRM	TSAVE
	TMK	LOC,K
	BLJ	CALLER
	TMR	TSAVE
	TMI	N
	AUM	N
	BOZ	RET
	SKI	
	BOZ	RET
ILOOP	TMA	LOC-1,K
	TAM	LOC,K
	TMA	COUNT-1,K
	TAM	COUNT,K
	SOK	1

22

BWI ILOOP
BUC RET

*
ITM HXI
NII
TMA N
IKA
CZM N
BOZ NEW
PLDOP CMA LOC,K
BOZ OLD
BWK FLOOP
NEW TMK N
SOK 1
TKM N
DLB TIM COUNT,K
TAM LOC,K
OUT RXI
BUC 0,J
ENDS

\$FILEMA
ESTAB,5,ITM,0,1,0,4
EXIT
\$EOJ

\$JOB FIO
\$OPTION 23
\$ASSIGN 6 0
\$ASSIGN 5 LR
\$REW,5
\$FORTRAN

```
SUBROUTINE SCRA(I,J)
  CALL QUIET(I)
  CALL IO('17+64*I,J,0)
  RETURN
END
SUBROUTINE OPN(LUN)
  CALL IO1(64*LUN+7)
  RETURN
END
SUBROUTINE CLOS(LUN)
  CALL IO1(LUN*64+8)
  RETURN
END
SUBROUTINE QUIET(LUN)
  CALL IO1(64*LUN)
  RETURN
END
SUBROUTINE REW(LUN)
  CALL QUIET(LUN)
  CALL IO1(64*LUN+'16)
  RETURN
END
SUBROUTINE WEF(LUN)
  CALL QUIET(LUN)
  CALL IO1(64*LUN+6)
  RETURN
END
SUBROUTINE RDAN(LUN;N,X)
  CALL QUIET(LUN)
  CALL IO(LUN*64+1,N,X)
  RETURN
END
SUBROUTINE WRAN(LUN;N,X)
  CALL QUIET(LUN)
  CALL IO(64*LUN+2,N,X)
  RETURN
END
SUBROUTINE READ(LUN;N,X)
  CALL QUIET(LUN)
  CALL IO(64*LUN+3,N,X)
  RETURN
END
SUBROUTINE WRIT(LUN;N,X)
  CALL QUIET(LUN)
  CALL IO(64*LUN+4,N,X)
  RETURN
END
SUBROUTINE RDANW(LUN,N,X)
  CALL RDAN(LUN,N,X)
  CALL QUIET(LUN)
```

```
RETURN
END
SUBROUTINE WRANW(LUN,N,X)
  CALL WRAN(LUN,N,X)
  CALL QUIET(LUN)
RETURN
END
SUBROUTINE READW(LUN,N,X)
  CALL READ(LUN,N,X)
  CALL QUIET(LUN)
RETURN
END
SUBROUTINE WRITW(LUN,N,X)
  CALL WRIT(LUN,N,X)
  CALL QUIET(LUN)
RETURN
END$
```

```
$FILEMA
ESTAB,5,F10,0,1,0,4
EXIT
$EOJ
```

\$JOB IO
\$ASSIGN,6,0
\$OPTION 23
\$ASSIGN 5 LR
\$REW,5
\$ASSEMB

NAME IO
IDEN IO
XDEF IO,IO
XDEF IO1,I
I GAP 1
TJM RET
TIM ISAVE
AGAIN TMI ISAVE
TMA 0,I
NAK
BLU \$I/O
BON AGAIN
BUC* RET
IO GAP 1
TMA 0,I
TAM PCKT
GAP 1
TMA 0,I
TAM PCKT+1
GAP 1
TIM PCKT+2
TJM RET
TLO PCKT
BLU \$I/O
BUC* RET
RET ***
PCKT BLOK 3
ISAVE ***
END\$

\$FILEMA
ESTAB,5,IO,0,1,0,4
EXIT
\$EOJ

26

\$JOB PROJS
\$ASSIGN 6 0
\$FILEMA
CREATE,PROJS,0,1,112,1
EXIT
\$EOJ

```

$JOB CSF
$OPTIONS 0 8 23
$ASSIGN 6 0
$OPTION 23
$ASSIGN 7 7
$ASSIGN 10,22,22,W2
$ASSIGN 5 LR
$ASSIGN 15 LR
$REW,5
$ASSEMBLER
    XDEF  SQ,SQ
    XDEF  IO,GQ
    XDEF  TQ,TQ
GO    TJM  RET
    GAP  1
    TMD  0,I
    TDM  STUFF
    TIK
    BLU  '35
STUFF ***
    ***
    BUC*  RET
    TO    GAP  1
    TJM  RET
    TIK
    BLU  '37
    BUC*  RET
    SQ    GAP  1
    TJM  RET
    TIK
    BLU  '36
    BUC*  RET
RET   ***
    ENDS
$FILEMA
ESTAB,5;CSF,0,1,0,4
EXIT
$EOJ

```

```
$JOB WAIT
$OPTION 23
$ASSIGN 6 0
$ASSIGN 5 LR
$REW,5
$ASSEMB
    IDEN WAIT
    XDEF WAIT,WAIT
WAIT   GAP 1
    TJM RET
    TIK
    BLU $WAI
    BUC* RET
RET    DATA 0
    ENDS
$FILEMA
ESTAB,5,WAIT,0,1,0,4
EXIT
$EOJ
```

```

$JOB SQUISH
$OPTION 23
$ASSIGN,6,0
$ASSIGN,5,LR
$REW,5
$FORTRAN
    SUBROUTINE SQUISH(IBUFF,N)
    DIMENSION IBUFF(135)
    DATA IR/'40/
    N=0
    J=0
1    J=J+1
    IF(J-135)2,100,100
2    N=N+1
    IBUFF(N)=IBUFF(J)
    IF(IBUFF(J)-18)1,9,1
9    IBUFF(N)=1
3    J=J+1
    IF(J-135)109,100,100
109  IF(IBUFF(J)-18)2,110,2
110  IBUFF(N)=IBUFF(N)+1
    IF(IBUFF(N)-'37)3;2;3
100  N=N-1
    IF(N.EQ.1)GOTO 101
    IF(IBUFF(N).GT.'40)GOTO 101
    GOTO 100
101  N=N+1
    IBUFF(N)=0
    RETURN
    ENDS
$FILEMA
ESTAB,5,SQUISH,0,1,0,4
EXIT
$EOJ
$EOJ

```

\$JOB ASR
\$ASSIGN 5 LR
\$REW,5
\$ASSIGN 6,0
\$OPTIONS 23
\$ASSEMB

IDEN ASR TTY HANDLER MODIFIED BY J BENSON
XDEF ASR,ASR
XDEF AFLAGS,AFLAGS
XDEF ASRIN,ASRIN
XDEF ASROUT,ASROUT
XDEF LFFL,LFFL
XDEF OCBUF,OCBUF
XDEF OCMSG,OCMSG
XDEF OCRDY,OCRDY
XDEF OCENTR,OCENTR

DISINT EQIV '200
TTY EQIV '0000

ASR TAE
ISW TTY
DOB 1
BOZ *-2

* PUNCT
TEA
BOZ \$CSRTN.
BNN NOTCLR
TSM \$ASRBSY
TSM AFLAGS
BUC 0,J

NOTCLR COB 1
BOZ READ
COB 2
BOZ WRITE
COB 6
BON FERR
COB '10
BOP FERR
BUC PENAB4

FERR TMA DISINT
UE1
BSL \$DEALRS
TOC '10
BUC 0,J
READ BLJ HOLDDTTY
TOA R0
DMA AFLAGS
TAM AFLAGS
TNA 3
TAM INBYT

TVV INXVXD
* WORD COUNT
TMK 3,K
TMA 1,K
TAM 1,I
CMA =24

31

BNP **2
 TOA 24
 NAA
 TAM INCNT
 TAM INCNT1
 * PILL BLANK S
 TOA INPNT
 TAM RSPNT
 TNI 24
 TMA BLANKS
 TAM INBUF+24, I
 BWI *-1
 TOB '212
 DDW TTY
 BNZ *-1
 TFM \$ASRBSY
 BLJ RELTTY
 TLO INCNT
 BSL \$WAITP

LINE FEED

* INPUT DONE
 TMA DISINT
 UI1
 TMJ 3,K
 TMJ 2,J
 TOI INBUF
 TMA 0,I
 TAM 0,J
 AOI 1
 AOJ 1
 AUM INCNT1
 BON *-5
 TMI 1,K
 TMI 1,I
 SOI 24
 BNP **6

NII
 TMA BLANKS
 TAM 0,J
 AOJ 1
 BWI *-2
 TMA DISINT

UE1
 BSL \$DEALRS

DEAL LOCATEP

TOC 4
 BUC 0,J
 BLJ HOLDTTY
 TMK 3,K
 TMA 1,K
 TAM 1,I
 EMA =24
 BNP **2
 TOA 24

NAA
 TAM OUTCNT
 TMJ 2,K
 TMI OUTBFD

WRITE

32

TAK
 TMA 0,J
 TAM 0,I
 AOI 1
 AOJ 1
 BWK *-4
 BSL KILBLNX
 TMA AFLAGS
 DOB R1
 TAM AFLAGS
 TNA 3
 TAM OUTBYT
 TMA OUTBFD
 TAM OUTPNT
 TNA 2
 TAM LFFL
 TMA* OUTPNT
 LRA 8
 KOB '61
 BOZ DOLF
 AUM LFFL
 KOB '60
 BOZ DOLF
 TZM LFFL
 KOB '40
 BOZ DOLF
 KOB '53
 BNZ SEND1
 LRA 8
 TAM* OUTPNT
 TFM OUTBYT
 BUC SENDIT
 BOLF TAM* OUTPNT
 AUM OUTBYT
 SEND1 TOB '212
 SENDIT DDW TTY
 BNZ *-1
 PENABL TFM \$ASRBSY
 TOA 5
 DCW TTY
 BNZ *-1
 PENAB4 BLJ RELTTY
 BSL \$DEALRS
 TOC 4
 BUC 0,J
 KILBLNX ***
 TMI OUTCNT
 PII
 TMA OUTBUF-1,I
 SMA BLANKS
 BNZ *+3
 AUM OUTCNT
 BUC KILBLNX+1
 BUC* KILBLNX
 OCMSG ***
 HXI
 TAM SAVA

33

TMA DISINT
 UI1
 RXI
 TMA SAVA
 BSL \$REAL*
 TME OCMMSG
 TMA ='177777

DAE
 TEM RETAD
 BSL \$DEALRS
 TRM MBUF+3
 TMI \$CAPLE
 TMD 1,I
 TDM MBUF
 TOA '272
 TAM MBUF+2

NII
 TIM \$OCUSER
 TLO MBUF
 BLU '37
 OCREJ BLJ RELTTY
 TMJ RETAD
 TZM \$OCUSER
 BUC 0,J

SAVA ***
 HOLDDTY HXI
 EZM \$ASRBSY
 BNN HOLDT1

HOLODT1 RXI
 TLO \$ASRBSY
 BSL \$WAITP
 BUC HOLDDTY
 TMA DISINT
 UI1
 TMA \$TTYIN
 UI1

RELTTY RXI
 BUC 0,J
 TMA \$TTYIN
 UE1
 TMA DISINT
 UE1

ASRIN BUC 0,J

TRM INSAVE
 IDW TTY
 BNZ *-1
 TAE
 KOB '377
 BOZ OCREJ
 KOB '337
 BOZ KEXIT
 TMA AFLAGS
 DOB R2
 BOP OCINPT
 TMA AFLAGS
 DOB R0

INTERRUPT INPUT RTINE

34

	BNZ	NKEXIT
	TEM*	\$MEMSIZ
	TME	'200
	TZA	
	TD4	
NKEXIT	BUC	KEXIT
	TEA	
	KOB	'336
	BOZ	RSETIN
	TFM	\$ASRBSY
	KOB	'212
	BOZ	KEXIT
	DDW	TTY
	BNZ	*-1
	KOB	'215
	BOZ	INFIN
PACKW	TMA	AFLAGS
	TOA	'177
	DAE	
	TMA*	INPNT
	TMJ	INBYT
	EXM	SHIFT1+3,i
	TEB	
	EXM	SHIFT2+3,i
	TAM*	INPNT
	XUM	INBYT
	BNZ	KEXIT
	TNA	3
	TAM	INBYT
	XUM	INPNT
	XUM	INCNT
	BON	KEXIT
	TFM	CRFL
INFIN	TOA	5
	DCW	TTY
	BNZ	*-1
	TMA	AFLAGS
	DOB	6
	TAM	AFLAGS
	TZM	INCNT
	TME	DISINT
	TZA	
	TD4	
KEXIT	TMR	INSAVE
	BRL*	ASRIN
*		
RSETIN	TNI	24
	TMA	BLANKS
	TAM	INBUF+24,i
	BWI	*-1
	TMA	INCNT1
	TAM	INCNT
	TNA	3
	TAM	INBYT
	TMA	INBUFD
	TAM	INPNT
	TFM	LFLL

35

TFM CRFL
TOA 5
DCW TTY
BNZ *-1
BUC KEXIT

*
OCREQ TMA AFLAGS
DOB R2
BNZ KEXIT
CZM OCRDY
BNZ KEXIT
AOM R2
DAC AFLAGS
TOA OCPNT
TAM RSPNT
TNI 24
TMA BLANKS
TAM OCBUF+24,I
BWI *-1
TMD OCN24
TDM OCCNT
TMA OCBUFD
TAM OCPNT
BUC KEXIT

OCINPT TEA
KOB '336
BOZ RSET2
KOB '215
BOZ OCFIN
KOB '212
BOZ KEXIT
TOA '177
DAE

TMA* OCPNT
TMI OCBYT
EXM SHIFT1+3,I
TEB
EXM SHIFT2+3,I
TAM* OCPNT
AUM OCBYT
BNZ KEXIT
TNA 3
TAM OCBYT
AUM OCPNT
AUM OCCNT
BON KEXIT

OCFIN TMA AFLAGS
DOB 3
TAM AFLAGS
TOA INPNT
TAM RSPNT
TFM OCRDY
TNI '22
BSL* PPPPP
BUC KEXIT
RSET2 TFM \$UPARF

36

	TFM	CRFL
	TMA	AFLAGS
	DOB	3
	TAM	AFLAGS
	BUC	KEXIT
OCENTR	HXI	
	TJM	RETOC
	TNI	24
	TMJ	OCBUFD
OCELP	TMA	0,K
	TAM	0,J
	AOK	1
	AQJ	1
	BWI	OCELP
	TFM	OCRDY
	TNI	'22
	BSL*	PPPPP
	TMJ	RETOC
	RXI	
	BUC	0,J
RETOC	***	
ASROUT	***	
	TRM	OUTSAV
	TZM	CRFL
	BNZ	DOCR
	AUM	LFLL
	BOP	NOTLF
	TOA	'212
	ODW	TTY
	BNZ	*-1
PEX	TMR	OUTSAV
	BRL*	ASROUT
NOTLF	TZM	LFLL
	TMA	AFLAGS
	DOB	R1
	BOZ	INHASR
	CZM	OUTCNT
	BOZ	OUTFIN
	TMA*	OUTPNT
	LRA	8
	DOB	'200
	DDW	TTY
	BNZ	*-1
	TAM*	OUTPNT
	AUM	OUTBYT
	BON	PEX
	AUM	OUTPNT
	AUM	OUTCNT
	TNA	3
	TAM	OUTBYT
	BUC	PEX
OUTFIN	TMA	AFLAGS
	DOB	5
	TAM	AFLAGS
	BOZ	*+2

	TFM	LFFL	
	TOA	'215	
	ODW	TTY	
	BNZ	*-1	
	BUC	PEX	
DOER	TOA	'215	
	ODW	TTY	
	BNZ	*-1	
	BUC	PEX	
INHASR	TMA	AFLAGS	
INH2	TZM	\$ASRBSY	
	TOA	4	
	OCW	TTY	
	BNZ	*-1	
	TME	DISINT	
	TZA		
	TD4		
	BUC	PEX	
AFLAGS	HLT		
INPNT	HLT		
INENT	HLT		
INBYT	HLT		
INBUFD	DAC	INBUF	
*			
OCPNT	HLT		
OCENT	HLT		
OCBYT	HLT		
OCBUFD	DAC	OCBUF	
*			
INBUF	BLOK	24	
BLANKS	DATA	'10020040	
DCBUF	RDAT	24('10020040)	
	DATA	'10020040	
P1	DATA	-1	
MBUF	RDAT	24('10020040)	
OCRDY	HLT		
OUTBFD	DAC	OUTBUF	
*			
CRFL	HLT		
LFFL	HLT		
HOLDLF	HLT		
INENT1	HLT		
SHIFT1	NOP		
	RRA	8	
	NOP		
SHIFT2	RRA	8	
	LRA	8	
	NOP		
OCN24	DATA	-24	
	DATA	-3	
BSHFT	RLA	8	
	NOP		
	RLA	16	
BSHIFT1	LRD	8	
	TEA		
	LRD	16	
BSPNT	DATA	0	

38

```
INSAVE  BLOK  5
OUTSAV  BLOK  5
OUTPNT  HLT
OUTCNT  HLT
OUTBYT  HLT
OUTBUF  BLOK  24
RETAD   ***
REGSAV  BLOK  5
B0      EQIV  1
B1      EQIV  2
B2      EQIV  4
B8      EQIV  '400
B9      EQIV  '1000
PPPPP   DAC   $REAL*,I
ENDS
```

```
$FILEMA
ESTAB,5,ASR,0,1,0,4
EXIT
$EOJ
```

\$JOB SIR
\$ASSIGN,6,0
\$ASSIGN,7,11
\$REW,7
\$ASSIGN,10,23,23,W1
\$REW,10
\$UTILITY
\$SRCEUD
\$REPL 21,56
\$INSE 91
SMA \$BUFSZ
\$REPL 115,110
\$REPL 122,123
TOA 1
BLJ \$ADWRIT
\$REPL 137,138
\$REPL 141
BLJ \$ADREAD
\$REPL 165,180
TOI 5
TMD PNAME
TNK '103
BSL* PPPP
\$INSE 181
PPPP DAC \$REAL*,K
PNAME DATA '22247111
DATA '25020040
\$INSE EOF
\$EXIT
\$ASSIGN,7,23
\$REW,7
\$OPTIONS
\$OPTIONS 23
\$ASSIGN,5,LR
\$REW,5
\$ASSEMB
\$FILEMA
ESTAB,5,\$IR,0,1,0,4
EXIT
\$EOJ

40

\$JOB MT
 \$ASSIGN 5 LR
 \$REW,5
 \$ASSIGN 6,0
 \$OPTIONS 23
 \$ASSEMBLER

IDEN MT HANDLER

ENTER HANDLER FROM IOC WITH

(I)≠FOB ADDRESS

(J)≠OPEN

(K)≠TEMP ADDRESS

(E)≠PDCT RELATIVE ENTRY ADDRESS

(A)≠FUNCTION CODE

(O,I)≠STATUS WORD

BIT 23 = BUFFER BUSY

22 = WORD COUNT NOT COMPLETE

21 = EOF

20 = ERROR

19 = (1) OPENED, (0) CLOSED

18 = (1) FOB, (0) DEVICE

17 = (1) PERMANENT ASSIGNMENT

12 - 16 = ERROR CODE (IF 20 SET)

6 - 11 = LOGICAL DEVICE NUMBER

0 - 5 = PHYSICAL DEVICE NUMBER

(1,I) = PARAMETER/PARAMETER ADDRESS

(1,K) = FOB ADDRESS

(2,K) = RETURN ADDRESS

(3,K) = USER PARAMETER/PARAMETER ADDRESS

(4-5,K) = USER E-A

RESET ENTRY:

(A)≠-1

(I)≠(DEVICE NO.) * (2)

(J)≠RETURN ADDRESS

XDEF MCAH,MCA0
 XDEF MCAH0,MCA0
 XDEF MCAH1,MCA1
 XDEF MCAH2,MCA2
 XDEF MCAIR,MCAI
 MCACU
 *
 MCA0
 MCA1
 MCA2
 MCST
 EQIV '0300

BLK MCST
 BLK MCST
 BLK MCST
 SOK MCA1
 CZA
 BOZ STATUS
 BON RESET
 TIM FCBP
 TKI
 TMK \$CURNT
 TMK 13,K
 TMJ 2,K
 TKM CW
 TMK 3,K
 BON *+3
 TME 2,K
 TMK 1,K

BUILD CORRECT TRANSPORT NO. IN (K)
 TEST FUNCTION CODE
 STATUS CHECK
 FUNCTION CODE =-1; RESET HANDLER

SET TRANSPORT NO. IN (I)
 RESTORE ADDRESS OF TALBE CONTANING TEMP
 ADDRESS OF TEMP
 PUT RETRUN ADDRESS IN (J)
 SAVE (K) TEMPORARILY
 PARAMETER ADDRESS
 SKIP ADDRESS AND WORD COUNT
 BUFFER ADDRESS IN (E)
 WORD COUNT IN (K)

41

TRM TRANS.
 COB '10
 BOZ CLOSE
 TMA \$DISINT
 UE1
 TMA FC
 TZM BFLAG
 TZM FOTFLAG
 BOZ MCST01
 TMD MESSX
 DIA
 TDM FOTMSG
 TMK FOTMAD
 BLU \$HOLD
 TMR TRANS.
 MCST01 TMK CW
 BSL \$DEAL
 COB MAXFC
 BNP MCST03
 MCST02 TOE '11
 BUC \$SABORT
 MCST03 TZM RETRYF
 TZM FRASEF
 TZM RPFLAG
 TOA 1
 TAM DFLAG
 TIA
 BSL ISW
 QBB R0
 BNZ MCST10
 TMA = '04010000
 BUC MCST05
 TMA = '04110000
 MCST05 AAM* FCBP
 BUC EXIT
 *
 MCST10 TMK FC
 QBB R1
 BOZ MCST20
 TMA FC
 COB '02
 BOZ MCST05-1
 COB '04
 BOZ MCST05-1
 COB '06 1
 BOZ MCST05-1
 MCST20 TOJ EXIT
 BUC* *,K
 DAC SR
 DAC SW
 DAC RR
 DAC RW
 DAC FRASE
 DAC WEOF
 DAC OPEN
 DAC CLOSE
 DAC RPF
 DAC RSF

SAVE REGISTERS
 IS IT A CLOSE

ALLOW DISPATCHER

RESTORE FUNCTION CODE
 CLEAR BINARY FLAG
 CLEAR EOT ELAG

OR IN TRANSPORT NO.
 STORE MESSAGE

HOLD-OUTPUT MESSAGE

RESTORE (K)
 DEALLOCATE CORE
 FUNCTION CODE TOO LARGE ?
 NO, GO ON
 ABORT CODE 11-ILLEGAL FUNCTION CODE
 ABORT ROUTINE
 CLEAR RETRY FLAG
 CLEAR ERASE FLAG
 CLEAR REPOSITION FLAG
 SET DIRECTION FLAG FORWARD

TRANSFER TRANSPORT NO. TO (A)
 INPUT HARDWARE STATUS WORD

BIT ZERO SET-DEVICE ON LINE
 ERROR CODE (1) DEVICE OFF-LINE

ERROR CODE(11) FILE PROTECTED
 MERGE WITH FCB WORD
 EXIT

RESTORE FC TO (K)
 QUERY BIT 1
 BIT 1 NOT SET = FILE NOT PROTECTED
 RESTORE FUNCTION CODE
 CHECK FOR SYMBOLIC WRITE
 FILE PROTECTED;CANNOT WRITE
 CHECK FOR BINARY WRITE
 FILE PROTECTED;CANNOT WRITE
 CHECK FOR WRITE EOF

BUILD EXIT

SYMBOLIC READ (01)
 SYMBOLIC WRITE (02)
 BINARY READ (03)
 BINARY WRITE (04)
 ERASE TAPE (05)
 WRITE END OF FILE (06)
 OPEN FILE (07)
 CLOSE FILE (10)
 REPOSITION FILE (11)
 BACKSPACE FILE (12)

42

	DAC	ADF	ADVANCE FILE (13)
	DAC	RSR	BACKSPACE RECORD (14)
	DAC	ADR	ADVANCE RECORD (15)
	DAC	REW	REWIND (16)
	DAC	SCRA	SET CRA (17)
	DAC	EXIT	SEEK CRA (ILLEGAL) (20)
	DAC	SCFA	SET CFA (21)
MAXFC	EQIV	*-MCST20-2	MAX. FUNCTION CODE
RESET	TOA	'1706	
	DCW*	MCACU	
	TMA	\$PDCTW2,I	
	BNN	*+3	
	NSA		
	TAM	\$PDCTW2,I	
	TMA	PIL	
	UA1		
	UE1		
	TMI	\$MCABF	
	BOZ	RESET1	IS IT ZERO
	NII		NEGATE ADDRESS
	TMA	0,I	NO-PICK UP 1ST WORD
	DMA	= '0140777Z	SAVE BITS 0-11,17,18
	TAM	0,I	RESTORE MODIFIED WORD
	TZM	\$MCABF	CLEAR BUSY FLAG
RESET1	BUC	0,J	EXIT
*			
STATUS	TKE		.SAVE TRANSPORT ,
	TMK	\$CURNT	.RESTORE TEMP TO (K)
	TMK	13,K	
	BLJ	\$CSREQ.	.GET FILE STATUS
*			RETURNS: (A)=STATUS (K)=USER RTN ADDRESS
*			(E)=UNCHANGED TRANSPORT .
	TEJ		.(J)=TRANSPORT .
	TME	CRA,J	.(E)=CRA
	EZA		.(C)=F(STATUS)
	BUC	0,K	.RETURN TO USER
*		BINARY/SYMBOLIC	READ /WRITE
BR	TME	RCODE	READ CODE
	TFM	RFLAG	SET BINARY FLAG
BSRW6	TMA	WC	PICK UP PARAMETER ADDRESS IF NEGATIVE
	BON	MCST02	IF NEG. ERROR
BSRW10	TNA	2	.RETRY COUNT
	TAM	RETRY.	SET FOR 5 TIME RETRY
BSRW12	TOA	WC	PICK UP ADDRESS OF WC
	DAW	MCACU	
	BNZ	*-1	BUSY
	TMA*	FCBP	PICK UP FCB WORD
	BON	*+3	BIT 23 SET ?
	NSA		NO
	TAM*	FCBP	RESET SIGN
BSRW15	TEA		(A) = COMMAND WORD
	TAM	CW	SAVE COMMAND WORD
BSRW18	TME	FCBP	FCB ADDRESS
	NEE		NEGATE
	TEM	\$MCABF	SET BUSY
BSRW20	DMA	\$MCATOT,I	MERGE TAPE OPTION WORD INTO COMMAND
	CZM	RFLAG	IS BINARY FLAG SET ?

43

BOZ **2
DMA = '77737777
OCW MCACU
BNZ *-1
BUC 0,J

RESET BINARY BIT

BUSY
EXIT (WITY (C)=0)

*
*
BW TME BINARY/SYMBOLIC WRITE
BUC WCODE
RSRW6-1

WRITE CODE
GO WRITE RECORD

*
*
SW TMA SYMBOLIC WRITE
LRA \$MCATOT,I
BNN 9

PICK UP TAPE OPTIONS WORD
TEST MODE BIT (14)
ASCII MODE; NO CONVERSION
CONVERT ASCII TO BCD
BASE ADDRESS
TEST TUPE BIT (16)

SW10 TMI SW20
TMK WC
QBB RA
BNZ R1
BLJ SW11
BUC \$S.CAE
SW12

9 TRACK CONVERT ASCII TO EBCD ✓ 3 CPW

SW11 BLJ \$S.CAB
SW12 TMI TRANS.
TMJ RA
SW20 TME WCODE
BUC RSRW6

7 TRACK CONVERT ASCII TO BCD ✓ 3 CPW
RESTORE TRANSPORT
RESTORE RETURN ADDRESS
WRITE CODE

*
*
SR TMA SYMBOLIC READ
LRA \$MCATOT,I
TAM 9

PICK UP TAPE OPTIONS WORD

TME ICFLAG
BLJ RCODE
TOK RSRW6
TMA ICFLAG
BNN 0,K

SET FLAG

READ RECORD

BSL SR30
TMI \$WAITP
TMK TRANS.
TMA RA
LLA \$MCATOT,I
BON 7

EXIT IF BINARY MODE
WAIT
RESTORE TRANSPORT

PICK UP TAPE OPTIONS WORD

SR10 TMI SR10
BLJ WC
BUC \$S.CEA
SR20

PICK UP WORD COUNT
CONVERT ASCII TO EBCD ✓ 3 CPW
EXIT
PICK UP WORD COUNT
CONVERT ASCII TO BCD ✓ 3 CPW

SR20 TMI \$S.CBA
TMA* FCBP
BOP **3
NSA

SR30 TAM* FCBP
TMJ RA
TZA
BUC 0,J

RESET BUFFER BUSY BIT

SET (C) = 0

*
WEOF TME WRITE END OF FILE
BUC FOF CODE
RSRW15

WRITE EOF CODE
GO WRITE EOF

44

* ERASE 3.5 * N IN. OF TAPE		
ERASE	TMA	WC PICK UP WORD COUNT
	BON	MCST02 ILLEGAL FUNCTION CODE
	NAA	
	TAM	FRASEF SET ERASE ELAG
	TOE	ERACODE ERASE CODE
	BUC	BSRW15 ACTIVATE FUNCTION
* OPEN		
	TZA	
	TAM	ERRCNT, I SET ERROR COUNT TO ZERO
	BUC	EXIT
* CLOSE		
	TMA*	FCBP PICK UP FIRST WORD
	DMA	= '01407777 SAVE BITS 4-11, 17, 18
	TAM*	FCBP RESTORE WORD
	TZM	FOTFLAG CLEAR EOT ELAG
	TME	ERRCNT, I PICK UP ERROR COUNT
	BOZ	CLOSE2
	RLD	3
	RLA	5
	RLD	3
	RLA	5
	RLD	3
		5
	ΔPA	= '14030060
	TME	= '10052060
	DIE	
	TAI	
	TMJ	= '10051105
	TMK	= '10047516
	TMA	RB
	BSL	\$OCMSGE
	TMK	CW
	BUC	\$CLOSEF
CLOSE2		REPOSITION FILE
* RPF		
	TMA	CFA, I
	BOZ	RPFE0F
RPF CFA	TAM	DRA
	BUC	SCRA
RPFE0F	TFM	RPFLAG
* BACKSPACE FILE		
BSP	TIA	
	BSL	ISW
	QBB	R6
	BNZ	FXI
	TOE	RSFTODE
	TFM	DFLAG
	BUC	BSRW15
* ADVANCE FILE		
ADF	TOE	ADFCODE
	BUC	BSRW15
* BACKSPACE RECORD		
BSR	TOE	BSRCODE
	TFM	DFLAG
		BACKSPACE RECORD CODE
		SET REVERSE FLAG

* BUC	RSRW15	ACTIVATE FUNCTION
* ADR	ADVANCE RECORD	
TOE	ADRCODE	ADVANCE RECORD CODE
BUC	RSRW15	ACTIVATE FUNCTION
* REW	REWIND	
TOA	REWCODE	REWIND CODE
TZM	\$MCABF	CLEAR BUSY FLAG
TZM*	EOF(I)	SET EOF =0
TZM*	CFA(I)	SET CFA=0
TZM*	CRA(I)	SET CRA=0
BUC	RSRW20	ACTIVATE FUNCTION
* SCRA	SET CURRENT RECORD ADDRESS	
TOA	'20	SET SC TO SEEK
TAM	FC	DROP INTO SEEK LOGIC
SEEK	TMA	COMPARE DESIRED ADDRESS WITH CURRENT ADDR
	CMA	
	CRA,I	
BON	RSR	
BOP	ADR	
BUC	0,J	CRA<DRA, ADVANCE RECORD
SCPA	TMA	
	CRA,I	
	TAM	
	CFA,I	
EXIT	TOC	SET RETURN C=ZERO
	BUC*	
	RA	
* ISW	INPUT HARDWARE STATUS	

LLA	3	POSITIONLN TRANSPORT NO.
DOB	6	OR DISABLE BITS
DCW	MCACU	
BNZ	*-1	BUSY RETURN
ISW	MCACU	
BNZ	*-1	BUSY RETURN
BUC*	ISW	EXIT-(A) CONTAINS STATUS
* MCAI	INTERRUPT PROCESSOR	

TRM	IPREG	ENTER VIA INTERRUPTED EXECUTED BSL
EZM	\$MCABF	SAVE ALL REGISTERS
BOZ	IPEXIT	WAS DEVICE BUSY?
TMI	TRANS.	
ISW	MCACU	(I)=TRANSPORT NO.
BNZ	*-1	
TAM	HSW	SAVE HARDWARE STATUS
QBB	R7	
BNZ	IPEXIT	
QBB	R2	TROUBLE
BNZ	TROUBLE	YES
MCAIJB	TMA	
	HSW	
QBB	R4	END OF TAPE?
BOZ	*+2	NO
TFM	EOTFLAG	SET EOT FLAG
AUM	FRASEF	
BOP	*+4	
TOA	ERACODE	SKIP ERASE
BLJ	RSRW18	(A) CONTAINS ERASE CODE
BUC	IPEXIT	EXECUTE ERASE
TZM	RETRYF	EXIT.
BNN	*+4	ZERO RETRY FLAG
		SKIP IF RETRY FLAG NOT SET

46

	TME	CW	
	BLJ	RSRW12	RESTART READ/WRITE
	BUC	IPEXIT	EXIT
*		BUILD HANDLER STATUS WORD	
	TMA	FC	PICK UP FUNCTION CODE
	TZE		NO
	COB	5	IS IT READ/WRITE
	BON	*+2	
	BUC	MCAI10	
	TMK	WC	WORD COUNT
	TMA	HSW	(A)=HARDWARE STATUS
	BNN	MCAI10	ABC IS NOT BUSY
*	ABC	NOT COMPLETE	
	JAW	MCACU	
	SMA	RA	
	TAK		SET DIFFERENCE IN (K)
	TOA	6	:DISABLE PT AND CLEAR ABC CHANNEL
	OCW*	MCACU	
	TME	=B22	SET BIT 22 FOR WC NOT COMPLETE
MCAI10	TMA	HSW	RESTORE HARDWARE STATUS
	QBB	R6	LOAD POINT?
	BNZ	MCAI20	YES, ZERO POINTERS
	QBB	R5	EOF?
	BOZ	MCAI30	NO-UPDATE CRA ONLY
	TZM	RPFLAG	WAS THIS AN RPE
	BNN	MCAI12	NO
	TZM	DFLAG	ZERO DIRECTION FLAG
	TOE	ADFCODE	ADVANCE OVER EOF
	BLJ	RSRW15	
	BUC	IPEXIT	EXIT
MCAI12	AME	=B21	SET EOF STATUS BIT
	TMA	DFLAG	UPDATE EOF
	AAM	FOF, I	
	BUC	MCAI21	
MCAI20	TZE		ZERO STATUS
	TZM*	FOF(I)	
MCAI21	TZM*	CFA(I)	ZERO CFA
	TZM*	CRA(I)	ZERO CRA
MCAI22	TMI	FCBP	
	TMA	0, I	
	BNN	*+2	SKIP IF NOT BUSY
	NSA		SET TO NOT BUSY
	DEA		
	TAM	0, I	
	TKM	1, I	STORE WORD COUNT
	BUC	MCAI31	
MCAI30	TMA	DFLAG	ADD OR SUBTRACT 1 FROM CRA
	AAM	CRA, I	
	BUC	MCAI22	
MCAI31	TMA	FC	TEST FUNCTION CODE
	COB	*20	SEEK IN PROGRESS
	BNZ	*+3	
	BLJ	SEEK	YES, CONTINUE SEEK
	BUC	IPEXIT	EXIT FROM INTERRUPT
MCAI135	NOP		
MCAI136	NOP		
EOREXIT	TZM	\$MCABF	CLEAR BUSY FLAG

.TRIGGER DISPATCHER (REF SYSDAT)

	TME	\$DISINT	
	TZA		
	TD4		
IPEXIT	TMR	IPREG	RESTORE REGISTERS
	TZM	ICFLAG	CLEAR FLAG FOR WAIT
	BRL*	MCAI	EXIT
		ERRORS	
* TROUBLE	TOA	6	
	DCW*	MCACU	
	AOM	1	INCREMENT ERROR COUNT
	DAC	FRRCNT,1	
	AUM	RETRY.	INCREMENT RETRY COUNTER
	BON	RETRY10	TRY AGAIN
* IF SS4 DO	NOT RETURN ERROR STATUS		
	QSS	R4	
	BNZ	MCAIJB	JUST IGNORE TROUBLE
	TMA	CW	IF B8=1, READ ERROR
	LLA	15	IF B8=0, WRITE ERROR
	BON	RETRY	READ ERROR - BACKSPACE RECORD-EXIT
	TMA	= '04030000	WRITE ERROR
	BUC	RETRY3	GO MERGE AND EXIT
RETRY	TOA	RSRCODE	BACKSPACE RECORD
	BLJ	RSRW18	
	TMA	= '04020000	
RETRY3	TME*	FCBP	READ ERROR
	NSE		STATUS WROB
	OAE		RESET SIGN
	TEM*	FCBP	
	BUC	FOREXIT	ERROR EXIT
RETRY10	TMA	CW	READ OR WRITE?
	LLA	15	B 8=1, READ/ B8=0, WRITE
	BON	RETRY30	READ
	TMA	RETRY.	
	AOA	1	
	BOZ	RETRY30	IF ZERO, 1ST TRY
	TFM	ERASEF	SET ERASE FLAG
RETRY30	TFM	RETRYF	SET RETRY FLAG
	TOA	RSRCODE	
	BLJ	RSRW18	
	BUC	IPEXIT	EXIT
* EOTFLAG	ZZZ	0	END OF TAPE FLAG
EOTMAD	'02	FOTMSG	
EOTMSG	DATA	'26054130	
	DATA	'26054130	
BFLAG	ZZZ	0	BINARY MODE FLAG
ICFLAG	ZZZ	0	SYMBOLIC READ FLAG
RETRYF	ZZZ	0	RETRY FLAG
ERASEF	ZZZ	0	ERASE FLAG
RPFLAG	ZZZ	0	REPOSITION FLAG
DFLAG	DATA	0	
RETRY.	DATA	0	
TRANS.	DATA	0	
RA	ZZZ	0	RETURN ADDRESS
WC	ZZZ	0	WORD COUNT
BA	ZZZ	0	BASE ADDRESS

48

```

FC      ZZZ      0
IPREG   BLOK     5
BB      DATA    '10020040
BRA     EQUV     WC
PIL     DATA    R13
FCBP    DATA    0
HSW     DATA    0
CW      DATA    0
RCODE   DATA    '40000503
WCODE   DATA    '60000103
EOFCODE EQUV     '203
ERACODE EQUV     '303
ERRCNT  RDAT     8(0)
ADFCODE EQUV     '703
BSPCODE EQUV     '1303
REWCODE EQUV     '402
ADRCODE EQUV     '603
BSRCODE EQUV     '1203
ABFLAG  EQUV     RA
ZC      EQUV     '4
CFA(I)  DAC      CFA,I
CFA      RDAT     8(0)
CRA(I)  DAC      CRA,I
CRA      RDAT     8(0)
EOF(I)  DAC      EOF,I
EOF      RDAT     8(0)
MESSX   DATA    '21247524
         DATA    '10052060
ENDS

```

FUNCTION CODE

FCB POINTER
HARDWARE STATUS WORD
COMMAND WORD
READ
WRITE

ZERO CONDITION BIT

CURRENT FILE ADDRESS TABLE

CURRENT RECORD ADDRESS TABLE
POINTER EOF TABLE
EOF NUMBER TABLE

```

$FILEMA
ESTAB,5,MT,0,1.0,4
EXIT
$EOJ

```

49

```

JOB PRNMES
ASSIGN,6,0
REW,5
INCLUDE CSF
FORTRA
    DIMENSION MES(8)
    DIMENSION MOUT(24)
    DATA MFS/6HPRNMES/
    CALL IQ(MES)
    ENCODE(72,222,MOUT)MES(1),MES(2)
222  FORMAT('LPR FIN',2I6)
    IF(MES(2).GT.1)CALL TQ(MOUT)
    CALL EXIT
    ENDS
CATALOG
TYPE=FG
NAME=PRNMES
BEGIN
EOJ

```


\$JOB SYSQ
\$ASSIGN,7,7,6,0
\$ASSIGN 5 LR
\$REW,5
\$OPTIONS 0 23
\$ASSEMBLE

IDEN SYSQ
XDEF SYSQ,SYSQ
XDEF CQUE,CQUE
XDEF COMM,COMM

WAIT EQIV '6
SFUNC EQIV '7
FROGS EQIV '11
DISINT EQIV '200
BSIZ EQIV 300
KEY1 DAC '77767
KEY2 DAC '77770
KEY3 DAC '77771
QW DATA 0
SYSQ TJM RETAD

* DONT START UP ANYTHING NEW IF FILE LOCK WORDS ARE SET

CZM* KEY1
BON SYSRET
CZM* KEY2
BON SYSRET
CZM* KEY3
BON SYSRET

SY\$LOP TOI QUE
TMD 0,1
CZE

BOZ SYSRET
TIM TEMP
BLU SFUNC
DATA 2
CME =-1
BOZ INIT
TMI TEMP
AOI 10
BUC SYSLOP

INIT TMI TEMP
TMD 0,1
TDM PNAME
TOA 127
TAM PRI
TMA PNAME

A AV
SNZ 1+2
AUM PRI
TMD PNAME
TMI PRI
BLU FROGS
DATA 1
TMI TEMP
AOI 10

SYSRET TMJ RETAD
BUC 0,J

NAV DATA '23440526

51

PNAME	DATA	0,0
RETAD	***	
PRI	***	
TEMP	***	
QUE	DATA	0
	BLOK	RSIZ
P	DATA	0
QWAIT	RXI	
	BSL	\$REAL*
	TFM	QW
	TLO	QW
	BSL	\$WAITP
	BSL	\$DEALRS
CQUE	HXI	
	TMI	P
	TOA	RSIZ
	SIA	
	BOZ	QWAIT
	BON	QWAIT
	TMD	0,K
	TDM	QUE,I
	AOK	2
	AOI	2
	TMD	0,K
	TDM	QUE,I
	AOK	2
	AOI	2
	TMD	0,K
	TDM	QUE,I
	AOK	2
	AOI	2
	TMD	0,K
	TDM	QUE,I
	AOK	2
	AOI	2
	TMD	0,K
	TDM	QUE,I
	AOK	2
	AOI	2
	TIM	P
	TOA	0
	TAM	QUE,I
	RXI	
	BUC	0,J
COMM	TOI	QUE
COMLOP	TMD	0,J
	CME	0,I
	BNZ	NXT
	EMA	1,I
	BOZ	HIT
NXT	TOA	0
	EMA	0,I
	ROZ	DONE
	AOI	10
	BUC	COMLOP
DONE	TMA	=-1
	BUC	2,J
HIT	AOI	2

52

TMD 0,I
 TDM 0,K
 AOI 2
 AOK 2
 TMD 0,I
 TDM 0,K
 AOI 2
 AOK 2
 TMD 0,I
 TDM 0,K
 AOI 2
 AOK 2
 TMD 0,I
 TDM 0,K
 AOI 2
 AOK 2
 TIK
 SOI 10
 TMA DISINT
 UI1
 TMD 0,K
 TDM 0,I
 CZE
 BOZ THRU -
 AOI 2
 AOK 2
 TMD 0,K
 TDM 0,I
 AOI 2
 AOK 2
 TMD 0,K
 TDM 0,I
 AOI 2
 AOK 2
 TMD 0,K
 TDM 0,I
 AOI 2
 AOK 2
 TMD 0,K
 TDM 0,I
 AOI 2
 AOK 2
 BUC POP10
 TMA P
 SOA 10
 TAM P
 TSM QW
 TMA DISINT
 UE1
 BUC 2,J
 ENDS

POP10

THRU

53

\$FILEMA
 ESTAB,5,SYSQ,0,1,0,4
 EXIT
 \$EO

```

$JOB PRINT
$ASSIGN,6,0
$OPTION 23
$REW,5
$ASSEMB

```

```

      IDEN  C
      XDEF  C,C
      GAP   1
      TJM   RET
      TIK
      BLU   '26
      BUC*   RET
RET    ***
START  BLL   $MAIN
      BUC   *-1
      ENDS  START

```

```

$FORTRA
      SUBROUTINE MAIN
      DIMENSION I(330),J(112),K(135),L(45)
      DIMENSION MES(10)
      DATA MES/6HPRNMES/
      LINE=0
      NCT=0
      CALL IO1(2*64+'7)
1      CALL C(J)
      NCT=NCT+1
      CALL CRACK(330,J(3);1)
      N=0
2      DO 10 JA=1,135
10     K(JA)='40
3      M=0
4      M=M+1
5      N=N+1
      IF(I(N).LT.'40)GOTO 9
      KTM)=I(N)
      GOTO 4
9      IF(I(N).EQ.0)GOTO 99
      IC=I(N)
      DO 90 JA=1,IC
      K(M)='40
90     M=M+1
      GOTO 5
99     CALL PACK(135,K,L)
      CALL IO(2*64+2,45;L)
      CALL IO1(2*64)
      LINE=LINE+1
      IF(I(N+1).NE.0)GOTO 2
      IF(J(2).EQ.0)GOTO 1
      CALL IO1(2*64+'10)
      MES(3)=LINE
      MES(4)=NCT
      CALL SQ(MES)
      RETURN
      ENDS

```

```

$INCLUDE STAGE
$INCLUDE IO
$INCLUDE CSF
$CATALOG

```

54

TYPE=FG
NAME=PRINT
ASSIGN 2=1
BEGIN
\$EOJ

55

\$JOB PR1
 \$OPTION 23
 \$ASSIGN 6 0
 \$ASSIGN,5,LR
 \$REW,5
 \$ASSEMBLER

	IDEN	PRINTER STAGEING ROUTINE
	XDEF	PR1,PR1
	XDEF	PRG,PRG
KEY	DATA	0
EOD	DATA	0
B	BLOK	110
E	BLOK	135
RUFF	BLOK	45
COMP	RDAT	330(0)
N	***	
M	DATA	0
BLANK	DATA	'10020040
ADDR	***	
RET	***	
REQ	***	
COUNT	***	
KSAVE	***	

PR1	CZA	
	BON	RESET
	BOZ	\$CSRIN.
	TAM	REQ
	COB	'10
	BOZ	CLOSE
	COB	'2
	BNZ	OUT
	TJM	RET
	TMK	3,K
	TMA	1,K
	TAM	COUNT
	TMA	2,K
	TAM	ADDR
	TMA	0,1
	BOP	*+3
	DMA	= '37777777
	TAM	0,1
	TMA	REQ
	COB	'2
OUT	BOZ	WRITE
	TMA	'200
	UE1	
	BSL	\$DEALRS
	TOC	'4
	BUC	0,J
WRITE	TMA	'200
	UE1	
	TNK	45
	TMA	BLANK
L1	TAM	RUFF+45,K
	BWK	1.1
	TMA	COUNT

56

	CMA	=45
	BNP	*+2
	TOA	45
	NAK	
	TOI	0
L2	TMJ	ADDR
	TMA	0,J
	TAM	RUFF,I
	AOJ	1
	AOI	1
	BWK	L2
	BLL	\$CRACK
	DAC	=135
	DAC	RUFF
	DAC	E
	BLL	\$SQUISH
	DAC	E
BACK	TMA	N
	AMA	M
	CMA	=328
	BOP	FLUSH
	TMK	M
	TAM	M
	TMI	N
	NII	
	TOJ	0
L5	TMA	E,J
	TAM	COMP,K
	AOJ	1
	AOK	1
	BWI	L5
FLUSH	BUC	OUT
	TLO	KEY
	BSL	\$WAITP
	TMK	M
	TOE	0
	TEM	COMP,K
	TZM	M
	BLL	\$PACK
	DAC	=330
	DAC	COMP
	DAC	R
	TMA	REQ
	COB	'10
	BNZ	NOCLOS
	TFM	FOD
	TFM	KEY
	TMK	KSAVE
	BUC	\$CLOSEF
NOCLOS	TZM	FOD
	TFM	KEY
	TZM	'77777
	TME	'200
	TZA	
	TD4	
	BUC	RACK

57

CLOSE	TMA	0,I
	DMA	= '01407777
	TAM	0,I
	TMA	'200
	UE1	
	TKM	KSAVE
	BUC	FLUSH
RESET	TZM	\$PR1F
	TMA	\$PDCTW2,I
	BNN	*+3
	NSA	
	TAM	\$PDCTW2,I
	BUC	0,J
*		
PRG	TLO	KEY
	BUC	0,J

\$

\$FILEMA END
 ESTAB,5,PR1,0,1,0,4
 EXIT

58

\$JOB OPCOM2
 \$ASSIGN 6 0
 \$OPTION 23
 \$REW,5
 \$INCLUDE LTV
 \$ASSEMB

	XDEF	TVIR,TVIR
CP	DAC	'77722
CL	***	
SAVE	RDAT	5(0)
TVIR	***	
	TRM	SAVE
	TMA*	CP
	RLA	12
	TAM	CL
	BLL	\$LTV
	DAC	CL
	TAM	CL
	TMA*	CP
	DMA	= '7777
	LLA	10
	DMA	CL
	TAE	
	TMA	= '1040
	DCW	'101
	BNZ	*-1
	TEA	
	ODW	'101
	BNZ	*-1
	TMR	SAVE
	BRL*	TVIR
	ENDS	

\$ASSEMB

	IDEN	CALLER
	XDEF	CALLER,CALLER
PRI	***	
PROG	***	

CALLER	GAP	1
	TMD	0,I
	TDM	PROG
	GAP	1
	TJM	RET
	TMI	0,I
	TIM	PRI
	BLU	\$SFUNC
	DATA	2
	TNA	1
	SAE	
	BNZ	DONE
	TMD	PROG
	TMI	PRI
	BLU	\$FROGS
	DATA	1
DONE	BUC*	RET
RET	***	

59

```

END
IDEN  RUMP
XDEF  RUMP,BUMP
XDEF  RUMP2,BUMP2
BUMP2 GAP 1
      TMA 0,I
      AOA 1
      CMA =499
      BNZ RET
      TOA 101
      BUC RET
BUMP  GAP 1
      TMA 0,I
      AOA 1
      CMA =100
      BNZ RET
      TOA 0
RET   TAM 0,I
      BUC 0,J
      END
IDEN  COMMUNICATIONS DRIVER FOR VARIOUS ASYNCHRONOUS BUFFERED TASKS
TTY   DATA '25052131
      DATA '10020040
TTYP  DATA 80
PRINT DATA '24051111
      DATA '23452040
PRINTP DATA 101
CNTRL DATA '20647124
      DATA '24446040
MES   ***
LNG   EQIV 24
IP    DATA '0217
IRN   DATA 0
      DATA 0
OP    DATA '0217
O N   DA A 0
R     DATA 0
IPK   DATA '0202
      DATA 24
      DAC 1B
OPK   DATA '0201
      DATA 24
      DAC 0B
IBF   DATA 0      -1 IMP INPUT BUFFER FULL
IDLE  EQIV '77777
GOT   DATA 0
IB    BLOK LNG
OBE   DATA -1      -1 IMP OUTPUT BUFFER EMETY
      DATA '10020040
OB    BLOK LNG
TTYB  DATA '0102
      DATA 24
      BAC 0B-1
ASRFLG DAC '1062
PIP   DATA '0217
      101
PIRN  DATA 0
POP   DATA '0217

```

60

PORN DATA 101
 DATA 0
 PIPK DATA '0202
 DATA 112
 PIBLOC ***
 POPK DATA '0201
 DATA 112
 DAC POB
 POB BLOK 112
 POBE DATA -1
 CNT DATA 90
 PCNT DATA 390

NET AVAILABLE TTY MESSAGES

*
 INTRTN TMA IDLE
 BOP INTEX
 TZM IDLE
 TME '200
 TZA
 TD4
 INTEX BUC 0,J

*
 WAITL CZM GOT
 BNZ ACTIV
 CZM IDLE
 BNN ACTIV

* 100 MILLISECOND WAIT

* (CALL TO INTERVAL TIMER MULTIPLEXOR)

TLO INTRTN
 TOI 100
 BLU '30
 TLO IDLE

* WAIT TILL ACTIVITY OR TIMEOUT RESETS IDLE FLAG
 BLU \$WAIT

* CALL TO SYSQ

ACTIV BLU '34
 TZM GOT

* INVOKE TTY IF CALLED FOR

TESTTY TMA IDLE
 KOB '212
 BNZ TWSC
 BLL \$CALLER
 DAC TTY
 DAC TTYP
 TZM IDLE
 TWSC CZM IDLE
 BOZ PINCK
 BON PINCK
 TMA IDLE
 TAM MES
 TLO CNTRL
 BLU '36

AUM GOT
 PINCK TLO PIB
 TFM IDLE
 BLU '27
 TKM PIBLOC
 TMA 0,K

61

BOZ PNOI
 * DONT PUT RECORD IN QUE IF IT IS FULL
 TMA PCNT
 SOA 1
 BON PNOI
 TAM PCNT

*
 TLO PIP
 BLU \$I/O
 TLO PIPK
 BLU \$I/O
 TNK '0200
 BLU \$I/O
 TZM* PIBLOC
 BLL \$BUMP2
 DAC PIRN
 AUM GOT
 PNOI CZM POBE
 BOZ INCK
 TMA PIRN
 SMA PORN
 BOZ INCK
 TLO POP
 BLU \$I/O
 TLO POPK
 BLU \$I/O
 TNK '0200
 BLU \$I/O
 BLL \$BUMP2
 DAC PORN
 TZM POBE
 AUM GOT
 INCK CZM IBF
 BOZ NOI

* INPUT BUFFER IS FILL -- EMPTY IT

TLO IP
 BLU \$I/O
 TLO IPK
 BLU \$I/O
 TNK '0200
 BLU \$I/O

* IRN=MON(IRN+1,100)

BLL \$BUMP
 DAC IRN
 TZM IBF
 AUM GOT

* WAIT IF OUTPUT BUFFER IS FULL

NOI CZM OBE
 BOZ CALTTO

* WAIT IF BUFFER IS EMPTY BUT NO RECORDS WAITING TO GO OUT

TMA IRN
 SMA ORN
 BOZ WAITL

* FILL OUTPUT BUFFER

TLO OP
 BLU \$I/O
 TLO OPK
 BLU \$I/O

62

	TNK	'0200
	BLU	\$I/O
	BLL	\$BUMP
	DAC	ORN
	TZM	OBE
CALTTO	TMA*	ASRFLG
	BON	WAITL
	TLO	TTYB
	BLU	\$I/O
	TFM	OBE
	AUM	CNT
	BUC	WAITL
* CODE LINKED BY BLU FROM PROGRAMS WHICH WILL FILL INPUT BUFFER		
PILLP	TJA	
	TKE	
	CZM	POBE
	BNN	**+3
	TLO	POBE
	BLU	\$WAIT
	HXI	
	TAJ	
	TEK	
	TMA	'200
	UI1	
	RXI	
	AUM	PCNT
	TNI	112
PLDOP	TMA	POB+112,I
	TAM	0,K
	AOK	1
	BWI	PLOOP
	TZM	IDLE
	TFM	POBE
	BUC	RET
PILLI	TJA	
	TKE	
* WAIT IF TOO MANY RECORDS IN QUF		
	CZM	CNT
	BNN	**+3
	TLO	CNT
	BLU	\$WAIT
* WAIT IF BUFFER ALREADY FULL		
	CZM	1BF
	BNN	**+3
	TLO	1BF
	BLU	\$WAIT
* FORBID DISPATCHER		
	HXI	
	TAJ	
	TEK	
	TMA	'200
	UI1	
	RXI	
	TMA	CNT
	SOA	1
	AM	CN

63

```

      TFM      IBX
      TZM      IDLE
* COPY USERS OUTPUT TO IB
      TNI      LNG
ILOOP  TMA      0,K
      TAM      IB+LNG,I
      AOK      1
      BWI      ILOOP
      BUC      RET
* ENABLE DISINT AND GO BACK
RET     TMA      '200
      1
      XVE      '200
      TZA
      TD4
      BUC      0,J
* PLUG LINKAGES INT BLU AREA AND DO FILE OPEN
START   TMA      INST1
      TAM*     L37
      TMA      INST3
      TAM*     L26
      TNK      '0107
      BLU      $I/O
      TNK      '0207
      BLU      $I/O
      HIT
      TOT      100
      RCT
      BLL      $CALLER
      DAC      PRINT
      DAC      PRINTP
MAIN    TMA      INST
      TAM*     INTRPT
      TMA      LVEL
      UA1
      UE1
      BUC      WAITL
LVEL    DATA   B17
INTRPT  DAC      '111
INST     RSL     $TVIR
INST1    BUL     FILLI
INST3    BUL     FILLP
L37      DAC     '37
L33      DAC     '33
L26      DAC     '26
      ENDS     START
$CATALOG
TYPE=RFG,PRIV
NAME=OPCOM,2
ASSIGN 1=1,2=OPMES
BE
GIN

```

64

\$JOB HIO
\$ASSIGN,6,0
\$ASSIGN,5,LR
\$REW,5
\$ASSEMB

IDEN HIO
XDEF HIO,HIO

* CALLING SEQUENCE CU, CW, DW

IOCW DCW '00
IODW ODW '00
HIO GAP 1
TMA 0,1
DMA IOCW
TAM PLUG1
TMA 0,1
DMA IODW
TAM PLUG2
GAP 1

HXI
TMA 0,1

* NEGATIVE CW IMP DONT SEND

BON **3

PLUG1 ***
RNZ *-1
GAP 1
TMA 0,1

PLUG2 ***
RNZ *-1

RXI

BUC 0,J

ENDS

\$FILEMA
ESTAR,5,HIO,0,1,0,4
EXIT
\$EOJ

65

```
$JOB AGE
$ASSIGN 5 LR
$REW,5
$ASSIGN,6,0
$FORTRAN
  SUBROUTINE AGE(N)
  COMMON/SYSCOM/NDAT(100)
  DO 1 J=1,8
  K=J+82
1  NDAT(K)=NDAT(K)+1
  K=82+N
  NDAT(K)=0
  RETURN
  ENDS
$FILEMA
ESTAB,5,AGE,0,1,0,4
EXIT
$EOJ
```

66

\$JOB GETGAM
\$ASSIGN,6,0
\$ASSIGN 5 LR
\$REW,5
\$FORTRA

```
SUBROUTINE GETGAM(ID,IT,BETA,BDOT)
  DIMENSION J(560)
  DIMENSION MER(24)
  COMMON/SYSCOM/NDAT(100)
  DATA MFR/30HNO GAMMA SHIFTS AVAILABLE
  DATA NSECT/-1/
  NSEC=5*MOD(ID,100)
  IF(NSEC.EQ.NSECT)GOTO 90
  IF(NDAT(92).LT.0)CALL WAIT(NDAT(92))
  NDAT(92)=-1
  CALL OPN(9)
  CALL SCRA(9,NSEC)
  CALL READW(9,560,J)
  CALL CLOS(9)
  NDAT(92)=0
90  NSECT=NSEC
  BETA=0.
  BDOT=0.
  IHIT=0
  DO 1 JA=1,560,7
  IF(J(JA).NE.ID)GOTO 1
  KIND=J(JA+6)/4096
  IF(KIND.NE.14)GOTO 1
  IHIT=IHIT+1
  IF(IHIT.GT.1)GO TO 80
  BETA=J(JA+2)
  BDDT=J(JA+3)
80  IF(J(JA+1).GT.IT)GO TO 1
  BETA=J(JA+2)
  BDOT=J(JA+3)
1  CONTINUE
  IF(IHIT.EQ.0)CALL TG(MER)
  BETA=BETA/100.
  BDDT=BDOT/100.
  RETURN
ENDS
```

\$FILEMA
ESTAB,5,GETGAM,0,1,0,4
EXIT
\$EOJ

67

```

$JOB TVSAT
$ASSIGN,6,0
$ASSIGN,5,LR
$REW,5
$OPTION 9
$FORTRAN

```

```

SUBROUTINE TVSAT(IF,ILT,IET,IL,IE,IT,ID)

```

```

C INPUTS

```

```

C IF -- FRAME NUMBER
C ILT -- TV CURSOR LINE
C IET -- TV CURSOR ELEMENT

```

```

C OUTPUTS

```

```

C IL --- SSCC LINE
C IE --- SSCC ELEMENT
C IT -- FRAME START TIME

```

```

C M1 IS REEL DEF
C REEL
C YYDDD
C HHMMSS
C LS
C ES
C M2 IS FRAME DEF
C FRAME
C REEL
C REC ST
C ELE ST
C TVL
C TVE
C MAG

```

```

DIMENSION M1(14),M2(14)
DIMENSION NER1(24),NER2(24)
COMMON/SYSCOM/NDAT(100)
DATA NFR1/30HFRAME NON-EXIST
DATA NFR2/30HTAPE NON-EXIST
CALL LOOK(3,IF,M2)
IF(M2(1).EQ.0.OR.M2(2).EQ.0)GO TO 800
CALL LOOK(2,M2(2);M1 )
IF(M1(1).EQ.0.OR.M1(2).EQ.0)GO TO 900

```

```

C YYDDD

```

```

ID=M1(2)

```

```

C HHMMSS

```

```

IT=M1(3)

```

```

C LINE

```

```

IL=ILT-M2(5)
IL=2*IL
ISIZ=M2(7)
LINST=M1(4)
LINST=LINST+M2(3)-1
MAG=1
IF(ISIZ.LT.116)MAG=3
IF(ISIZ.LT.56)MAG=6
IF(ISIZ.GT.115)MAG=1
IL=IL/MAG+LINST

```

68

```

C ELEM(VTT=M1(5)-1+M2(4)
  IE=IET-M2(6)
  IE=6*IF
  IE=IE/MAG+IEST
  RETURN
800  CALL TO(NER1)
     GOTO 1000
900  CALL TO(NER2)
1000 IL=0
     IE=0
     IT=0
     ID=0
     RETURN
     END
  SUBROUTINE LOOK(LUN,KEY,MES)
C LOOKUP AND SAVE
C   LUN=2 == REELS
C   LUN=3 == FRAMES
  DIMENSION MES(14);M(300),ILOC(3)
  COMMON/SYSCOM/NDAT(100)
  DATA ILOC/0,94,93/
  DATA N/0/
  IF(NDAT(54).NE.0)N=0
  IF(N.EQ.0)GO TO 100
  DO 1 J=1,N,15
  IF(M(J+1).NE.KEY)GO TO 1
  IF(M(J).NE.LUN)GO TO 1
  GO TO 200
1   CONTINUE
100  IF(N.EQ.300)N=0
     M(N+1)=LUN
     IF(LUN.NE.2.AND.LUN.NE.3)STOP TVSAT
     I=ILOC(LUN)
     IF(NDAT(I).LT.0)CALL WAIT(NDAT(I))
     NDAT(I)=-1
     CALL LOOKUP(LUN,KEY;M(N+2))
     NDAT(I)=0
     J=N+1
     N=N+15
200  DO 201 I=1,14
     J=J+1
201  MES(I)=M(J)
     RETURN
  ENDS
$FILEMA
ESTAR,5,TVSAT,0,1,0,4
EXIT
$EOJ

```

69

```

$JOB LOOKUP
$ASSIGN 6 0
$OPTIONS 8 23
$ASSIGN 5 LR
$REW,5
$FORTRAN

```

```

C SUBROUTINE LOOKUP(LUN,KEY,MUNG)
  FOR LOOKING UP AN ENTRY IN FRAMES OR REELS FILE
  DIMENSION MUNG(14)
  COMMON/LOKP/NSEC,MUN,JSEC(112)
  CALL OPN(LUN)
  CALL REW(LUN)
  NSEC=-1
  MUN=LUN

```

```

C NUMBER OF ENTRIES IN FILE

```

```

N=JARY(1)-1
C DISTANCE FROM START OF FILE TO FIRST DATA ENTRY
IOFF=JARY(2)
DO 1 J=1,14
1  MUNG(J)=0
DO 2 J=10,N
  IF(JARY(J).EQ.KEY)GO TO 3
2  CONTINUE
  GO TO 100
3  ISEC=IOFF+J/8
  IREL=1+14*MOD(J,8)
  CALL SCRA(LUN,ISEC)
  CALL READW(LUN,112,JSEC)
  LST=IREL+13
  JA=0
  DO 4 J=IREL,LST
    JA=JA+1
4  MUNG(JA)=JSEC(J)
100 CALL CLOS(LUN)
  RETURN
  END

```

```

FUNCTION JARY(J)
C FUNCTION JARY (J) IS THE JTH ENTRY IN THE FILE INDEX
C PERFORMS IO IF NECESSARY

```

```

COMMON/LOKP/NSEC,MUN,JSEC(112)
MSEC=(J-1)/112
MWOR=MOD(J-1,112)+1
IF(NSEC.EQ.MSEC)GO TO 1
NSEC=MSEC
CALL SCRA(MUN,MSEC)
CALL READW(MUN,112,JSEC)
1 JARY=JSEC(MWOR)
  RETURN
  ENDS

```

```

$FILEMA
ESTAR,5,LOOKUP,0,1,0,4
EXIT
$EOJ

```

70

```
$JOB VERS
$REW,5
$INCLUDE IO
$INCLUDE FIO
$ASSIGN 6 0
$FORTRAN
    DIMENSION J(112)
    DATA J/112*0/
1    FORMAT(10A3)
    READ(7,1)(J(L),L=1,10)
    CALL OPN(2)
    CALL SCRA(2,0)
    CALL WRITW(2,112,J)
    CALL CLOS(2)
    CALL EXIT
    ENDS
$FILEMA
EXIT
$ASSIGN 2=VERS
$CATGO
MCIDAS VERS S2.10H1.00
$EOJ
```

```

$JOB TTY
$ASSIGN 6,0
$REW,5
$INCLUDE,CSF
$INCLUDE,STAGE
$INCLUDE,IO
$FORTRAN

```

```

SUBROUTINE QUITTS
COMMON/SYSCOM/NDAT(100)
CALL EXIT
RETURN
END

```

```

SUBROUTINE SQUASH(IS,ID)
DIMENSION IS(24),ID(8)
CALL PACK(24,IS,ID)
RETURN
END

```

```

SUBROUTINE MAKE(I,M,N,L)
DIMENSION M(24)
DATA MIN/'55/
DATA ICOM/'40/
N=0
L=0
ISGN=1

```

```

1 IF(I.GT.60)GO TO 100
  IF(M(I).EQ.MIN)GO TO 900
  IF(M(I).EQ.ICOM)GO TO 100
  KK=M(I)-48
  IF(KK.LT.0.OR.KK.GT.9)L=1
  N=10*N+KK
  I=I+1
  GO TO 1
100 I=I+1
    N=N+ISGN
101 RETURN
900 I=I+1
    IF(N.NE.0)L=1
    ISGN=-1
    GO TO 1
END

```

```

DIMENSION NE5(24)
DIMENSION LIS(300)
DIMENSION MES(27);M(60)
DIMENSION NERN(2)
DIMENSION NER(2)
COMMON/SYSCOM/NCOM(100)
DATA NE5/30HMUST LOGGIN FIRST
DATA NER/6HERROR /
DATA NERN/6HCRAP: /
DATA ISTAR/3H***/
DATA ICOM/3H.../
DATA IRLK/'40/
CALL CRACK(1,ISTAR,ISTAR)
CALL CRACK(1,ICOM,ICOM)
111 CALL IO1(64*1+'7)
    CALL IO(64*1+1,27;MES)

```

72

```

CALL IO1(64*1)
CALL IO1(64*1+'10)
CALL CRACK(60,MES,M)
CALL IO1(64*2+'7)
CALL IO(64*2+'17,0,0)
CALL IO(64*2+1,300,LIS )
CALL IO1(64*2+'10)
KD=0
DO 20 J=1,300,3
  IF(MES(1).NE.LIS(J))GO TO 20
  KO=J
  M1=LIS(J+1)
  M2=LIS(J+2)
20  CONTINUE
  IF(KO)302,302,301
301  CONTINUE
C  FORCES LOGGIN PROCEDURE
  IF(NCOM(1).EQ.0.AND.KO.NE.1)CALL TO(NE5)
  IF(NCOM(1).EQ.0.AND.KO.NE.1)CALL QUIT$
  IF(M(3).FQ.ISTAR)GO TO 100
  IF(M(3).FQ.IBLK) GO TO 200
300  CONTINUE
  MES(1)=NFR(1)
  MES(2)=NFR(2)
303  CONTINUE
  CALL SQUASH(M,MES(3))
  CALL TO(MES)
  CALL QUIT$
302  CALL OC(MES)
  CALL QUIT$
100  MES(1)=M1
  MES(2)=M2
  CALL SQUASH(M(4),MES(3))
  CALL SQ(MES)
  CALL QUIT$
200  MES(1)=M1
  MES(2)=M2
  I=2
  DO 201 J=3,10
  CALL MAKE(I,M(3),N,E)
  IF(L.NF,0)GO TO 300
201  MES(J)=N
  CALL SQ(MES)
  CALL QUIT$
  END$
$ASSEMB
XDEF  OC,OC
OC    GAP    1
      TJM    RET
      TIK
      BLU    '25
      BUC*   RET
RET   ***
      END$
$CATALOG
TYPE=FG
NAME=TTY
$ASSIGN. 1=1,2=NAMLIS
NOMAP
BEGIN
$EOJ

```

\$JOB NRZL
\$ASSIGN,5,LR
\$ASSIGN,6,0
\$REW,5
\$FORTRAN

```
SUBROUTINE NRZDEC(LINE,L,IFERROR)
  DIMENSION LINE(2);IB(96),I(28)
  DATA KS,LEVELB,LEVELT,ILOOK/-10,75,100,85/
  IB(N)=IIR(N)
  LASBIT=0
  IERROR=0
```

C KLUDGE TO TRY TO GET MORE DECODES
E CANT USE MORE THAN LINE AND TIME THEN

```
ILOOK=72
DO 2 I=1,ILOOK
  I2=2*I
  I1=I2-1
  N=LINE(I1)
  M=LINE(I2)
  NN=1
  MM=1
  IF(N.LT.LEVELT)NN=0
  IF(M.LT.LEVELT)MM=0
  IF(N.GT.LEVELB.AND.N.LT.LEVELT)NN=MM
  IF(M.GT.LEVELB.AND.M.LT.LEVELT)MM=NN
  IF(MM.EQ.NN)GO TO 1
  IERROR=1
RETURN
```

1 J=1
IF(LASBIT.EQ.MM)J=0

IB(I)=J
LASBIT=MM

2 CONTINUE

```
L(1)=2*IB(35+KS)+IB(36+KS)
L(2)=8*IB(37+KS)+4*IB(38+KS)+2*IB(39+KS)+IB(40+KS)
L(3)=8*IB(41+KS)+4*IB(42+KS)+2*IB(43+KS)+IB(44+KS)
L(4)=8*IB(45+KS)+4*IB(46+KS)+2*IB(47+KS)+IB(48+KS)
L(5)=2*IB(51+KS)+IB(52+KS)
L(6)=8*IB(53+KS)+4*IB(54+KS)+2*IB(55+KS)+IB(56+KS)
L(7)=4*IB(58+KS)+2*IB(59+KS)+IB(60+KS)
L(8)=8*IB(61+KS)+4*IB(62+KS)+2*IB(63+KS)+IB(64+KS)
L(9)=4*IB(66+KS)+2*IB(67+KS)+IB(68+KS)
L(10)=8*IB(69+KS)+4*IB(70+KS)+2*IB(71+KS)+IB(72+KS)
L(11)=4*IB(75+KS)+2*IB(76+KS)+IB(77+KS)
L(12)=2*IB(81+KS)+IB(82+KS)
L(13)=16*IB(81+KS)+8*IB(82+KS)+4*IB(83+KS)+2*IB(84+KS)+IB(85+KS)
L(14)=4*IB(88+KS)+2*IB(89+KS)+IB(90+KS)
L(15)=2*IB(91+KS)+IB(92+KS)
L(16)=16*IB(86+KS)+8*IB(87+KS)+4*IB(88+KS)+2*IB(89+KS)+IB(90+KS)
L(17)=4*IB(91+KS)+2*IB(92+KS)+IB(93+KS)
L(18)=2*IB(94+KS)+IB(95+KS)
L(19)=16*IB(91+KS)+8*IB(92+KS)+4*IB(93+KS)+2*IB(94+KS)+IB(95+KS)
L(20)=IB(97+KS)
L(21)=IB(98+KS)
L(22)=IB(99+KS)
L(23)=IB(100+KS)
```

74


```

L(24)=4*IB(18+KS)+2*IB(19+KS)+IB(20+KS)
L(25)=8*IB(21+KS)+4*IB(22+KS)+2*IB(23+KS)+IB(24+KS)
L(26)=IB(73+KS)
L(27)=IB(74+KS)
L(28)=IB(65+KS)
RETURN
5ND
SUBROUTINE N9ZL(LINE,ITIME)
DIMENSION J(100),K(300),L(28)
CALL -FADW(2,100,J)
CALL CRACK(300,J,K)
CALL NRZDEC(K,L,IE)

```

8

```

LINE=1000*L(1)+100*L(2)+10*L(3)+L(4)

```

8

```

90 ITIME=10*ITIME+L(JJ)
RETURN
1 LINE=0
ITIME=0
RETURN
ENDS

```

```

$FILEMA
ESTAB,9,NRZL,0,1,0,4
EXIT
$EOJ

```

75

```
$JOB ZLANDM
$REW,5
$ASSIGN,6,0
$ASSIGN 2=LANDMA
$INCLUDE FIO
$INCLUDE IO
$FORTRA
    DIMENSION JDIR(560)
    DATA JDIR/560*0/
    CALL OPN(2)
    N=0
    DO 1 J=1,100
    CALL SCRA(2,N)
    CALL WRITW(2,560,JDIR)
1   N=N+5
    CALL EXIT
    ENDS
$CATGO
$EOJ
```

\$JOB KLUDGE
\$ASSIGN 6 0
\$REW,5
\$FORTRAN

```
SUBROUTINE DOIT(LUN)
  DIMENSION J(112)
  CALL OPN(LUN)
  CALL REW(LUN)
  CALL READW(LUN,112,J)
  J(2)=10
  CALL SCRA(LUN,0)
  CALL WRITW(LUN,112,J)
  CALL CLOS(LUN)
  RETURN
END
DO 10 K=2,3
10  CALL DOIT(K)
  CALL EXIT
ENDS
```

\$INCLUDE IO
\$INCLUDE FIO
ASSIGN 2=REELS,3=FRAMES
\$CATGO
\$ASSIGN 6,1
\$EOJ

77

```

$JOB FRAMES
$OPTIONS 0 8 23
$REW,5
$ASSIGN 6,0
$INCLUDE IO
$INCLUDE FIO
$INCLUDE CSF
$FORTRAN
    DIMENSION J(1120)
    DO 10 K=1,1120
10      J(K)=0
        1 =10
        T-LM OPN(2)
           2)
        CALL M-M-W(2,1120;J)
        CALL CLOS(2)
        CALL EXIT
        ENDS
$ASSIGN 2 FRAMES
$CATGO
$EOJ

```

```
$JOB REELS
$OPTIONS 0 8 23
$REW,5
$ASSIGN 6,0
$INCLUDE IO
$INCLUDE FIO
$INCLUDE CSF
$FORTRAN
    DIMENSION J(1120)
    DO 10 K=1,1120
10      J(K)=0
        J(1)=10
        CALL OPN(2)
        CALL RFW(2)
        CALL WRITW(2,1120;J)
        CALL CLOS(2)
        CALL EXIT
    ENDS
$ASSIGN 2 REFLS
$CATGO
$EOJ
```

```
$JOB SETKEY
$ASSIGN,6,0
$REW,5
$INCLUDE CSF
$FORTRAN
    DIMENSION MES(10)
    COMMON/SYSCOM/NDAT(100)
    DATA MFS/6HSETKEY/
    CALL IO(MES)
    KEY=MES(1)
    KSTAT=MES(2)
    IF(KSTAT.NE.1.AND.KSTAT.NE.0)CALL ABORT
    IF(KEY.LT.0.OR.KEY.GT.22)CALL ABORT
    N=1
    IF(KEY.GT.0)N=2**KEY
    M=*77777777.XOR.N
    NDAT(95)=NDAT(95).AND.M
    N=N*KSTAT
    NDAT(95)=NDAT(95).OR.N
    CALL EXIT
END$
$CATALOG
TYPE=FG
NAME=SETKEY
BEGIN
$EOJ
```

80

```

$JOB DPROJ
$ASSIGN 6 0
$INCLUDE FIO
$INCLUDE IO
$FORTRAN
    DIMENSION N(112)
    DATA N/112*0/
    CALL OPN(2)
    CALL SCRA(2,0)
    I=0
1   FORMAT(15,1X,08)
10  READ(7,1)I1,I2
    IF(I1.EQ.9999)GO TO 100
    I=I+1
    J=2*(I-1)      +1
    N(U)=I1
    N(U+1)=I2
    GO TO 10
100 CALL WRITW(2,112,N)
    CALL EXIT
    ENDS

```

\$ASSIGN 2 PROJS

\$CATGO

```

1000 00000002
1210 00000002
1220 00000002
1230 00000002
1250 00000002
1500 00000002
1510 00000000
1520 00000002
1600 00000002
1610 00000002
1620 00000002
1630 00000002
1640 00000002
1800 00000002
1900 00000002
3710 00000002
4439 00000002
5600 00000002
   00 00000002
6700 00000002
6900 00000002
6999 40000000
9999 00000000

```

\$EOJ

81

```

$JOB DEFTAP
$REW,5
$INCLUDE CSF
$INCLUDE FIO
$INCLUDE IO
$ASSIGN 6 0
$INCLUDE WAIT
$FORTRAN

```

```

        DIMENSION NAM(2)
        DIMENSION JDIR(1120)
        DIMENSION MES(8),JSEC(112)
        DIMENSION MER(24),NER(24)
        COMMON/SYSCOM/IP(100)
        DATA MFR/30HNOT AUTHORIZED TO DT
        DATA NFR/30HTAPE REEL FILE FULL
        DATA NAM/6HDEFTAP/
        CALL WAIT(IP(94))
        IP(94)=-1
        CALL OPN(2)
199    CALL RFW(2)
        CALL RFADW(2,1120;JDIR)
1    MES(1)=NAM(1)
        MES(2)=NAM(2)
        CALL IO(MES)
        IKEY=IP(95).AND.'40000000
        IF(IKEY.NE.0)GO TO 6999
        CALL CLOS(2)
        IP(94)=0
        CALL TO(MER)
        CALL EXIT
6999  IF(NAM(1).EQ.MES(1))GO TO 299
        IF(JDIR(1).EQ.1000)GO TO 999
        KEY=KFIND(JDIR,MES(1))
            2 10+
        CALL ,GZF\(:,11K#ZT#E)
        IST=1+14*MOD(KEY,8)
        LST=IST+7
        DO 99 JAY=IST,LST
            INDX=1+JAY-IST
99    JSEC(JAY)=MES(INDX)
        CALL SCRA(2,10+KEY/8)
        CALL WRITW(2,112,JSEC)
100  CALL SCRA(2,0)
        CALL WRITW(2,1120;JDIR)
        GO TO 199
299  CALL CLOS(2)
        IP(54)=1
        IP(94)=0
        CALL EXIT
999  CALL TO(NER)
        GO TO 299
        END
        FUNCTION KFIND(J,K)
        DIMENSION J(1120)

```

82


```

      IF(J(1).EQ.10)GO TO 100
      N=J(1)-1
      DO 2 L=10,N
      IF(J(L).EQ.K)GO TO 200
2      CONTINUE
100    KFOUND=J(1)
      J(KFOUND)=K
      J(1)=J(1)+1
      RETURN
200    KFOUND=L
      RETURN
      ENDS

```

```

$CATALOG
TYPE=FG
NAME=DEFTAP
$ASSIGN 2=REELS
BEGIN
$EOJ

```

83

```

$JOB DFRAME
$ASSIGN 6 0
$REW,5
$INCLUDE CSF
$INCLUDE FIO
$INCLUDE IO
$INCLUDE WAIT
$FORTRAN
    DIMENSION NAM(2)
    DIMENSION JDIR(1120)
    DIMENSION MES(8),JSEC(112)
    DIMENSION NER(24),NERR(24)
    COMMON/SYSCOM/NDAT(100)
    DATA NAM/6HDFRAME/
    DATA NERR/30HERROR -- FRAME LOCK
    DATA NER/30HFRAME FILE FULL
    CALL WAIT(NDAT(93))
    NDAT(93)=-1
    CALL OPN(2)
199  CALL REW(2)
    CALL READW(2,1120,JDIR)
1  MES(1)=NAM(1)
   MES(2)=NAM(2)
   CALL IO(MES)
   IF(NAM(1).EQ.MES(1))GO TO 299
   IF(JDIR(1).EQ.1000)GO TO 999
   KEY=KFIND(JDIR,MES(1))
   CALL SCRA(2,10+KEY/8)
   CALL READW(2,112,JSEC)
   IST=1+14*MOD(KEY,8)
   LST=IST+7
   NP=JSEC(IST+13)
C PRIVILEGED PROJECT
   IKEY=NDAT(95).AND.'40000000
   IF(IKEY.NE.0)GO TO 98
C UNRESTRICTED FRAME
   IF(NP.EQ.0)GO TO 98
C PROJECT MATCH
   IF(NP.EQ.IPROJ)GO TO 98
   CALL TO(NERR)
   GO TO 199
98  DO 99 JAY=IST,LST
     INDX=1+JAY-IST
99  JSEC(JAY)=MES(INDX)
     CALL SCRA(2,10+KEY/8)
     CALL WRITW(2,112,JSEC)
180 CALL SCRA(2,0)
     CALL WRITW(2,1120,JDIR)
     GO TO 199
299 CALL CLOS(2)
C SIGNAL SYSTEM CHANGE WORD
   NDAT(54)=1
C RELEASE FRAME LOCK
   NDAT(93)=0

```

84

```

CALL EXIT
999 CALL TO(NER)
GO TO 299
END
FUNCTION KFIND(J,K)
DIMENSION J(1120)
IF(J(1).EQ.10)GO TO 100
N=J(1)-1
DO 2 L=10,N
IF(J(L).EQ.K)GO TO 200
2 CONTINUE
100 KFIND=J(1)
J(KFIND)=K
J(1)=J(1)+1
RETURN
200 KFIND=I
RETURN
END$

```

```

$CATALOG
TYPE=FG
NAME=DFRAME
ASSIGN 2=FRAMES
BEGIN
$EOJ

```

```

$JOB WHATAP
$REW,5
$INCLUDE CSF
$INCLUDE IO
$INCLUDE FIO
$INCLUDE WAIT
$ASSIGN,6,0
$FORTRAN
    DIMENSION JDIR(1120,MES(8),MOUT(24)
    COMMON/TLIS/NLIS,NDAY(1000)
    COMMON/SYSCOM/NDAT(100)
    DATA MFS/6HWHATAP/,
    NLIS=0
    CALL IO(MES)
    CALL WAIT(NDAT(94))
    NDAT(94)=-1
    CALL OPN(2)
    CALL REW(2)
    CALL READW(2,1120;JDIR)
    N=JDIR(1)-1
    NSEC=JDIR(2)
    DO 100 J=10,N,80
    CALL SCRA(2,NSEC+J/8)
    CALL READW(2,1120;JDIR)
    DO 100 K=1,80
    L=2+(K-1)*14
100  CALL Q(JDIR(L))
    CALL CLOS(2)
    NDAT(94)=0
    DO 200 J=1,NLIS
    ENCODE(72,222,MOUT)NDAY(J)
200  CALL TO(MOUT)
    CALL EXIT
222  FORMAT(I10)
    END
    SUBROUTINE Q(N)
    COMMON/TLIS/NLIS,NDAY(1000)
    IF(N.EQ.0)RETURN
    IF(NLIS.NE.0)GO TO 1
    NLIS=1
    NDAY(1)=N
1    DO 2 J=1,NLIS
    IF(NDAY(J).EQ.N)RETURN
    IF(NDAY(J).GT.N)GO TO 3
2    CONTINUE
    NLIS=NLIS+1
    NDAY(NLIS)=N
    RETURN
3    DO 4 K=J,NLIS
    L=NLIS+J-K
4    NDAY(L+1)=NDAY(L)
    NLIS=NLIS+1
    NDAY(J)=N
    RETURN
END$

```

86

```

$CATALOG
TYPE=FG,PRIV
NAME=WHATAP
ASSIGN 2=REELS
BEGIN
$EOJ
$REW,5
$INCLUDE FIO
$INCLUDE IO
$INCLUDE CSF
$ASSIGN 6 0
$FORTRAN
    DIMENSION J(112),MES(10)
    COMMON/SYSCOM/NDAT(100)
    DATA MFS/6HDVERS /
    CALL OPN(2)
    CALL SCRA(2,0)
    CALL READW(2,112,J)
    CALL SCRA(2,0)
    CALL IO(MES)
    DO 1 K=1,8
1      J(K)=MFS(K)
    IF(NDAT(1).NE.152Q)CALL ABORT
    CALL WRITW(2,112,J)
    CALL CLOS(2)
    CALL EXIT
    ENDS
$CATALOG
TYPE=FG
NAME=DVERS
ASSIGN 2=VERS
BEGIN
$EOJ

```

```

$JOB RESFRM
$ASSIGN 6 0
$REW,5
$INCLUDE CSF
$INCLUDE FIO
$INCLUDE IO
$INCLUDE WAIT
$FORTRAN
    DIMENSION NAM(2)
    DIMENSION JDIR(1120)
    DIMENSION MES(8),JSEC(112)
    DIMENSION NERR(24)
    COMMON/SYSCOM/NDAT(100)
    DATA NAM/6HRESFRM/
    DATA NERR/30HERROR -- FRAME LOCK
    CALL WAIT(NDAT(93))
    NDAT(93)=-1
    CALL OPN(2)
199  CALL REW(2)
    CALL READW(2,1120,JDIR)
1  MES(1)=NAM(1)
    MES(2)=NAM(2)
    CALL IO(MES)
    IF(NAM(1).EQ.MES(1))GO TO 299
    IKEY=NDAT(95).AND.'40000000
    IF(IKEY.NE.0)GO TO 98
    CALL TO(NERR)
    GO TO 199
98  CONTINUE
    I1=MES(1)
    I2=MES(2)
        DO 777 JA=I1,I2
    KEY=KFIND(JDIR,JA)
    CALL SCRA(2,10+KEY/8)
    CALL READW(2,112,JSEC)
    CALL SCRA(2,10+KEY/8)
    IST=1+14*MOD(KEY,8)
    JSEC(IST)=MES(3)
    777 CALL WRITW(2,112,JSEC)
100  CALL SCRA(2,0)
    CALL WRITW(2,1120,JDIR)
    GO TO 199
299  CALL CLOS(2)
    NDAT(93)=0
    CALL EXIT
    END
    FUNCTION KFIND(J,K)
    DIMENSION J(1120)
    IF(J(1).FQ.10)GO TO 100
    N=J(1)-1
    DO 2 L=10,N
    IF(J(L).FQ.K)GO TO 200
2  CONTINUE
100  KFIND=J(1)
    J(KFIND)=K
    J(1)=J(1)+1

```

88

```
      RETURN  
280  KFINL=L  
      RETURN  
      ENDS
```

```
$CATALOG  
TYPE=FG  
NAME=RESFRM  
ASSIGN 2=FRAMES  
BEGIN  
$EOJ
```

```

$JOB LOGDUM
$REW,5
$INCLUDE IO
$ASSIGN 6 0
$INCLUDE FIO
$INCLUDE CSF
$FORTRAN
    DIMENSION MES(112),MOUT(24)
    DATA MES/6HLOGDUM/
    CALL IQ(MES)
    IF(MES(1).NE.987898)CALL EXIT
    CALL OPN(2)
    CALL RFW(2)
    CALL READW(2,112,MES)
    DO 10 J=1,112,2
    IF(MES(J).EQ.0)GO TO 10
    F=MES(J+1)
    F=F/60.
    ENCODE(72,222,MOUT)MES(J),F
222  FORMAT('PROJECT',I5,' ',F8.2,' MINUTES')
    CALL TQ(MOUT)
10   CONTINUE
    DO 11 J=1,112
11   MES(J)=0
    CALL SCRA(2,0)
    CALL WRITW(2,112,MES)
    CALL CLOS(2)
    CALL EXIT
    ENDS
$CATALOG
TYPE=FG
NAME=LOGDUM
ASSIGN 2=LOG
BEGIN
$EOJ

```



```
$JOB PRNDUT
$REW,5
$ASSIGN 6,0
$INCLUDE CSF
$FORTRAN
    DIMENSION MES(10);NAM(2)
    DIMENSION MOUT(24)
    DATA NAM/6HPRNDUT/
1    MES(1)=NAM(1)
    MES(2)=NAM(2)
    CALL IQ(MES)
    IF(MES(1).EQ.NAM(1))CALL EXIT
    J=MES(1)
    J=J+1
    ENCODE(72,222,MOUT)(MES(K),K=2,J)
    CALL TQ(MOUT)
    GO TO 1
222  FORMAT(7I8)
    ENDS
$CATALOG
NAME=PRNDUT
TYPE=FG
BEGIN
$EOJ
```

91

28

```
$JOB ERADIR
$REW,5
$INCLUDE CSF
$INCL DE IO
$INCLUDE FIO
$ASSIGN 6,0
$FORTRA
    DIMENSION MES(10),J(112)
    DATA MFS/6HERADIR/
    DATA J/112*0/
    CALL IO(MES)
    CALL OPN(2)
    CALL REW(2)
    CALL WRITW(2,112,J)
    CALL CLOS(2)
    CALL EXIT
    ENDS
$CATALOG
TYPE=FG
NAME=ERADIR
$ASSIGN 2=DKDIR
BEGIN
$EOJ
```

```

$JOB NRZDEC
$ASSIGN,6,0
$REW,5
$INCLUDE IO
$INCLUDE FIO
$INCLUDE STAGE
$INCLUDE CSF
$INCLUDE NRZL
$INCLUDE FTIME
$INCLUDE ITIME
$FORTRA
    FUNCTION IDEC(N)
    DATA I7/'60/
    NN=N
    N1=MOD(N,10)
    NN=NN/10
    N2=MOD(NN,10)
    NN=NN/10
    N3=MOD(NN,10)
    N1=N1+1Z
    N2=N2+1Z
    N3=N3+1Z
    N2=N2.SHIFT.8
    N3=N3.SHIFT.16
    IDEC=N1.OR.N2.OR.N3
    RETURN
    END
    DIMENSION M1(24),M2(24)
    DIMENSION MES(8)
    COMMON/SYSCOM/ICOM(100)
    DATA M1/30HTAPE START      TIME
    DATA M2/30H      PERCENT SUCCESSFUL DECODES
    DATA MES/6HNRZDEC/
    CALL IQ(MES)
    CALL OPN(2)
    CALL RFW(2)
    IG=0
    ID=-1
    LIMIT=MES(1)
    IF(LIMIT.GT.2000)LIMIT=2000
    DO 1 M=1,LIMIT
    IF(ICOM(1).EQ.0)GOTO 222
    CALL NRZL(LINE,ITM)
    IF(LINE.EQ.0)GOTO 1
    IG=IG+1
    NID=LINE+1-M
    IF(ID.EQ.NID)GOTO 1
    ID=NID
    T=FTIME(ITM)
    X=.6*FLOAT(LINE)/3600.
    T=T-X
    ITI=ITIME(T)
    M1(5)=IDEC(NID)
    M1(10)=IDEC(ITI)
    M1(9)=IDEC(ITI/1000)
    CALL TO(M1)

```

93

```

1      CONTINUE
      Y=LIMIT
      X=IG
      X=X/Y
      IX=100*X
      M2(1)=IDFC(IX)
      CALL TQ(M2)
222    CALL RFW(2)
      CALL CLOS(2)
      CALL EXIT
200    FORMAT('FRACTION OF SUCCESSFUL DECODES = ',E5,3)
      ENDS

```

```

$CATALOG
TYPE=FG
NAME=NRZDEC
ASSIGN 2=11
BEGIN
$EOJ

```

```

$JOB ACQTAP
$ASSIGN,6,0
$REW,5
$INCLUDE FIO
$INCLUDE IO
$INCLUDE CSF
$INCLUDE NRZI
$INCLUDE STAGE
$INCLUDE FTIME
$INCLUDE ITIME
$FORTRA
    DIMENSION MES(10),MOUT(10),MER(24)
    COMMON/SYSCOM/ICOM(100)
    DATA MES/6HACQTAP/
    DATA MOUT/6HDEFTAP/
    DATA MER/30HNO SUCCESSFUL DECODES
    CALL IO(MES)
    CALL OPN(2)
    CALL REW(2)
    DO 1 J=1,1000
    IF(ICOM(1).EQ.0)GOTO 222
    CALL NRZI(L,IT)
    IF(L.EQ.0)GOTO 1
    T=FTIME(IT)
    X=.6*FLOAT(L)/3600.
    T=T-X
    MOUT(3)=MES(1)
    MOUT(4)=MES(2)
    MOUT(5)=ITIME(T)
    MOUT(6)=L-J+1
    MOUT(7)=1
    MOUT(8)=0
    MOUT(9)=0
    MOUT(10)=0
    CALL SQ(MOUT)
    ENCODE(72,100,MER,MOUT)
    GOTO 2
1    CONTINUE
2    CALL TO(MER)
222  CALL REW(2)
    CALL CLOS(2)
    CALL EXIT
100  FORMAT(2A3,8I7)
    ENDS
$CATALO
NAME=ACQTAP
TYPE=FG
ASSIGN 2=11
BEGIN
EOJ

```

95

\$JOB STATUS
\$ASSIGN 6,0
\$REW,5
\$INCLUDE CSF
\$FORTRA

```
FUNCTION IOCT(N)
  DATA M1,M2,M3/'7,170,'700/
  DATA I7/3H000/
  N1=M1.AND.N
  N2=M2.AND.N
  N3=M3.AND.N
  N2=N2.SHIFT.5
  N3=N3.SHIFT.10
  IOCT=I7.OR.N1.OR.N2.OR.N3
  RETURN
END
```

```
SUBROUTINE MAIN(N)
  DIMENSION N(2)
  DIMENSION MES(10)
  DATA MES/6HSTATUS/
  CALL IO(MES)
  CALL PRINT(N(1))
  M=1
1  MM=II(N(17),M)
  IF(MM.EQ.0)RETURN
  MM=MM+16
  CALL PRINT(N(MM))
  M=MM
  GO TO 1
END
```

```
FUNCTION II(N,M)
  STARTS AT LOC N LOOKING FOR WORD M
  DIMENSION N(2)
  I=1
10  IF(N(I).EQ.M+128)GOTO 100
  IF(I.GT.500)GOTO 101
  I=I+8
  GO TO 10
100 II=I
  RETURN
101 II=0
  RETURN
END
```

```
SUBROUTINE PRINT(N)
  DIMENSION N(8)
  DIMENSION MOUT(24)
  DATA I7/3H /
  DATA N2/3H 00/
  MOUT(1)=N(2)
  MOUT(2)=N(3)
  DO 1 J=3,24
1  MOUT(J)=IB
  J=N(4).SHIFT.-18
  J1=J.AND.7
  J2=J.AND.'70
  J2=J2.SHIFT.5
  J=N2.OR.J1.OR.J2
```

96

```
MOUT(4)=J
MOUT(5)=IOCT(N(4)/512)
MOUT(6)=IOCT(N(4))
MOUT(8)=IOCT(N(6)/512)
MOUT(9)=IOCT(N(6))
NNN=N(7)
MOUT(11)=IOCT(NNN/512)
MOUT(12)=IOCT(NNN)
MOUT(14)=IOCT(N(8))
CALL TO(MOUT)
RETURN
END$
```

\$ASSEMB

```
START   BLL    $MAIN
        DAC    '201
        BLU    $EXI
        END$   START
```

\$CATALOG

```
TYPE=FG,PRIV
NAME=STATUS
BEGIN
$EDJ
```

```
$JOB SETF
$ASSIGN,6,0
$REW,5
$INCLUDE CSF
$FORTRAN
    DIMENSION MES(8)
    COMMON/SYSCOM/NDAT(100)
    DATA MFS/6HSETF /
    CALL IQ(MES)-
    J=NDAT(56).AND.'32770000
    J=J+MES(1)
    NDAT(56)=J
    CALL EXIT
    ENDS
$CATALOG
TYPE=FG
NAME=SETF
BEGIN
$EOJ
```

```
$JOB DLIM
$ASSIGN,6,0
$REW,5
$INCLUDE CSF
$FORTRAN
    DIMENSION MES(8)
    COMMON/SYSCOM/NDAT(100)
    DATA MES/6HDLIM /
    CALL IQ(MES)
C LOWER LIMIT+4096*LPPER LIMIT
    NDAT(57)=MES(1)+4096*MES(2)
    CALL EXIT
ENDS
$CATALOG
TYPE=FG
NAME=DLIM
BEGIN
$EOJ
```

```

$JOB DRATE
$ASSIGN,6,0
$REW,5
$INCLUDE CSF
$FORTRA
    DIMENSION MES(8)
    COMMON/SYSCOM/NDAT(100)
    DATA MFS/6HDRATE /
    CALL IO(MES)
C    ARGUMENT IS NUMBER OF FRMS PER SECOND TO LOG
    IF(MES(1).GT.30)CALL EXIT
    IF(MES(1).LE.0)CALL EXIT
    X=MES(1)
    X=1./X
    X=X*1000.
    NDAT(58)=X
    CALL EXIT
    ENDS
$CATALOG
TYPE=FG
NAME=DRATE
BEGIN
$EOJ

```

100

```

$JOB LOOP
$ASSIGN,6,0
$REW,5
$INCLUDE CSF
$FORTRA
    DIMENSION MES(8),NAM(2)
    COMMON/SYSCOM/NDAT(100)
    DATA NAM/6HLOOP /
    DATA MES/6HLOOP /
    CALL IQ(MES)
    IF(NDAT(1).EQ.0)CALL EXIT
1   MES(1)=NAM(1)
    MES(2)=NAM(2)
    CALL IQ(MES)
    IF(MES(1).NE.NAM(1))CALL EXIT
    IF(NDAT(1).EQ.0)CALL EXIT
    CALL SLEEP(NDAT(58))
    IT=NDAT(57)/4096
    IB=NDAT(57)-4096*IT
    ID=MOD(NDAT(56),4096)
    L=IT-IB
    IF(ID.EQ.IT)GOTO 2
    NDAT(56)=NDAT(56)+1
    GOTO 1
2   NDAT(56)=NDAT(56)*L
    GOTO 1
END$

```

```

$ASSEMBR
    IDEN  SLEEE
    XDEF  SLEEP,SLEEP
SLEEP   GAP  1
        TJM  RET
        TMI  0,I
        TLO  RTN
        BLU  '30
        TFM  K
        TLO  K
        BLU  $WAI
        BUC* RET
RET      ***
RTN      TZM  K
        TME  '200
        TZA
        TD4
        BUC  0,J
K        ***
        ENDS

```

```

$CATALOG
TYPE=FG
NAME=LOOP
BEGIN
$EOJ
$JOB NAVFRM
$ASSIGN,6,0
$REW,5
$INCLUDE CSF
$INCLUDE FIO
$INCLUDE IO

```

101

```
$INCLUDE LOOKUP
$INCLUDE TVSAT
$INCLUDE WAIT
$FORTRA
C CALLS NAVIGATION SYSTEM FOR DAY OF CURRENT FRAME-----
  DIMENSION MIN(8),MOUT(10)
  COMMON/SYSCOM/NDAT(100)
  DATA MIN/6HNAVFRM/
  DATA MOUT/6HNAVSYS/
  CALL IQ(MIN)
  IF=NDAT(56).AND.'Z777
  CALL TVSAT(IF,100,100,1,J,IT,ID)
  MOUT(3)=ID
  CALL SQ(MOUT)
  CALL EXIT
  ENDS
$CATALOG
TYPE=FG
NAME=NAVFRM
ASSIGN 2=REELS,3=FRAMES
BEGIN
$EOJ
```

102

\$JOB DEFPNT
\$ASSIGN,6,0
\$REW,5
\$INCLUDE CSF
\$FORTRA

DIMENSION MES(8)
COMMON/SYSCOM/NDAT(100)
DATA MES/6HDEFPT/
CALL IO(MES)
NDAT(59)=MES(1)
NDAT(60)=MES(2)
CALL EXIT
END\$

\$CATALOG
TYPE=FG
NAME=DEFPT
BEGIN
\$EOJ

103

```
$JOB DORBIT
$REW,5
$ASSIGN,6,0
$INCLUDE CSF
$FORTRA
    DIMENSION MIN(8),MOUT(10)
    DATA MIN/6HDORBIT/
    DATA MOUT/6HDLANDM/
C ORBITS ARE ENCODED AS 'LANDMARKS' OF TYPE 12 AND 13
    CALL IQ(MIN)
    MOUT(3)=MIN(1)
    MOUT(4)=0
    MOUT(5)=0
    MOUT(8)=MIN(2)
    MOUT(9)=MIN(3)
    MOUT(10)=12
    MOUT(6)=MIN(4)
    MOUT(7)=MIN(5)
    CALL SQ(MOUT)
    MOUT(10)=13
    MOUT(8)=MIN(6)
    MOUT(9)=MIN(7)
    MOUT(6)=MIN(8)
    MOUT(7)=0
    CALL SQ(MOUT)
    CALL EXIT
END$
$CATALOG
TYPE=FG
NAME=DORBIT
BEGIN
$EOJ
```

104

```

$JOB DSRATE
$ASSIGN,6,0
$REW,5
$INCLUDE CSF
$FORTRA
    DIMENSION MIN(8),MOUT(10)
    DATA MIN/6HDSRATE/
    DATA MOUT/6HDLANDM/
@ SPIN RATE IS ENCODED AS 'LANDMARK' OF KIND 11
    CALL IO(MIN)
    MOUT(3)=MIN(1)
    MOUT(4)=0
    MOUT(5)=0
    MOUT(6)=0
    MOUT(7)=0
    MOUT(8)=MIN(2)
    MOUT(9)=0
    MOUT(10)=11
    CALL SQ(MOUT)
    CALL EXIT
END$
$CATALOG
TYPE=FG
NAME=DSRATE
BEGIN
$EOJ

```

105

```

$JOB DLANDM
$ASSIGN 6 0
$REW,5
$INCLUDE CSF
$INCLUDE IO
$INCLUDE FIO
$INCLUDE WAIT
$FORTRA

```

```

SUBROUTINE SRT(J)
  DIMENSION J(560)
  DO 1 K=1,560,7
  DO 1 L=K,560,7
  JL=J(L+1)
  JK=J(K+1)
  IF(JL.FQ.0)JL=240100
  IF(JK.FQ.0)JK=240100
  IF(JK.LE.JL)GOTO 1
  DO 2 M=1,7
  MK=M-1+K
  ML=M-1+L
  LTEM=J(MK)
  J(MK)=J(ML)
  J(ML)=LTEM
2  CONTINUE
1  CONTINUE
  RETURN
  END

```

```

  DIMENSION NAM(2)
  DIMENSION MES(8)
  DIMENSION JDIR(560)
  DIMENSION NER3(24),NER9(24)
  COMMON/SYSCOM/NDAT(100)

```

```

C DEFINES LANDMARKS AND OTHER PICTURE RELATED DATA
C INPUT IS SYDDDD HHMMSS KEY P1,P2,P3,P4, KIND
C ONLY ONE ENTRY OF UNIQUE SYDDDD HHMMSS KIND
C FOR EXAMPLE, FOR LANDMARKS ...

```

```

C INPUT IS SYDDDD, HHMMSS, KEY, SATLIN, SATELE, LAT, LON
  DATA NER9/40HPICTURE INFO FILE UPDATE REJECTED

```

```

  DATA NAM/6HDLANDMZ
  DATA NER3/40HNO ROOM IN PICTURE DATA FILE

```

```

199  MES(1)=NAM(1)
     MES(2)=NAM(2)
     CALL IQ(MES)
     IF(MES(1).EQ.NAM(1))GO TO 299
     IF(MES(1).EQ.0)CALL ABORT
     IKEY=NDAT(95).AND.'40000000
     IF(IKEY.NE.0)GO TO 191
     CALL TO(NER9)
     CALL EXIT
191  CONTINUE
     ID=MES(1)
     IT=MES(2)
     IK=MES(3)
     IL=MES(4)
     IE=MES(5)
     ILAT=MES(6)

```

106


```

      ILON=MFS(7)
      KIND=MFS(8)
      CALL WAIT(NDAT(92))
      NDAT(92)=-1
      CALL OPN(4)
      MSEC=5*MOD(ID,100)
      CALL SCRA(4,MSEC)
      CALL READW(4,560,JDIR)
      DO 11 K=1,560,7
      IF(JDIR(K).NE.ID)GOTO 11
      IF(JDIR(K+1).NE.IT)GOTO 11
      KAY=JDIR(K+6)/4096
      IF(KAY.NE.KIND)GOTO 11
      GOTO 50
11  CONTINUE
      DO 10 K=1,560,7
      IF(JDIR(K).EQ.0)GOTO 50
10  CONTINUE
      CALL TO(NER3)
      CALL CLOS(4)
      NDAT(92)=0
      CALL EXIT
C SYDDDD
50  JDIR(K)=ID
      K=K+1
C HHMMSS
60  JDIR(K)=IT
      K=K+1
C LAT
      IF(ILAT.NE.0)JDIR(K)=ILAT
      IF(KIND.GT.9)JDIR(K)=ILAT
      K=K+1
C LONG
      IF(ILON.NE.0)JDIR(K)=ILON
      IF(KIND.GT.9)JDIR(K)=ILON
      K=K+1
C LINE
      IF(IL.NE.0)JDIR(K)=IL
      IF(KIND.GT.9)JDIR(K)=IL
      K=K+1
      IF(IE.NE.0)JDIR(K)=IE
      IF(KIND.GT.9)JDIR(K)=IE
      K=K+1
      JDIR(K)=IK+4096*KIND
C
      CALL SRT(JDIR)
      CALL SCRA(4,MSEC)
      CALL WRITW(4,560,JDIR)
      CALL CLOS(4)
      NDAT(92)=0
      GO TO 199
299 CALL EXIT
      ENDS

```

```

$CATALOG
TYPE=FG
NAME=DLANDM
$SSIGN 4=LANDMA
BEGIN
$EOJ

```

107

```

$JOB FINDTP
$ASSIGN 6,0
$INCLUDE IO
$INCLUDE FIO
$INCLUDE CSF
$INCLUDE WAIT
$FORTRAN
    DIMENSION JDIR(1120),MFS(8),MOUT(24)
    DIMENSION NAM(2)
    DIMENSION MTY(24)
    COMMON/SYSCOM/NDAT(100)
    DATA NAM/6HFINDTP/
    DATA MOUT/6HLISTAP/
    DATA MTY/30HNO TARE REELS
    CALL WAIT(NDAT(94))
    NDAT(94)=-1
    CALL OPN(2)
199  MES(1)=NAM(1)
    MES(2)=NAM(2)
    CALL IO(MES)
    IF(MES(1).EQ.NAM(1))GO TO 299
    CALL RFW(2)
    KN=0
    CALL READW(2,1120;JDIR)
    N=JDIR(1)-1
    IF(MES(1).EQ.0)GO TO 299
    DO 100 J=10,N,80
    CALL SCRA(2,10+J/8)
    CALL READW(2,1120;JDIR)
    DO 100 K=1,80
    L=1+(K-1)*14
    IF(JDIR(L+1).NE.MES(1))GO TO 100
    MOUT(3)=JDIR(L)
    DO 90 L=4,10
90  MOUT(L)=0
    KN=KN+1
    CALL SQ(MOUT)
100  CONTINUE
    IF(KN.EQ.0)CALL TQ(MTY)
    GO TO 199
299  CALL CLOS(2)
    NDAT(94)=0
    CALL EXIT
    ENDS
$CATALOG
TYPE=FG
NAME=FINDTP
ASSIGN 2=REELS
BEGIN
$EOJ

```

108

```

$JOB LISTAP
$REW,5
$ASSIGN 6,0
$INCLUDE LOOKUP
$INCLUDE,FIO
$INCLUDE IO
$INCLUDE CSF
$INCLUDE WAIT
$FORTRAN

```

```

      DIMENSION MES(8),MOUT(24),NAM(2),JSEC(14)
      COMMON/SYSCOM/NDAT(100)
      DATA NAM/6HLISTAP/
1      MES(1)=NAM(1)
      MES(2)=NAM(2)
      CALL IO(MES)
      IF(MES(1).EQ.NAM(1))CALL EXIT
      DO 10 J=1,8
      IF(MES(J).EQ.0)GO TO 10
      IF(NDAT(94).LT.0)CALL WAIT(NDAT(94))
      NDAT(94)=-1
      CALL LOOKUP(2,MES(J),JSEC)
      NDAT(94)=0
      IF(JSEC(1).EQ.0)GO TO 9
222  ENCODE(72,222,MOUT)(JSEC(K),K=1,5)
      FORMAT('T',2I7,I8,2I6)
      CALL TO(MOUT)
      GO TO 10
9      ENCODE(72,333,MOUT)MES(J)
333  FORMAT('NO TAPE',I7)
      CALL TO(MOUT)
10     CONTINUE
      GO TO 1
      ENDS

```

```

$CATALOG
TYPE=FG
NAME=LISTAP
ASSIGN 2=REELS
BEGIN
$EOJ

```

```

$JOB FINFRM
$ASSIGN 6 0
$REW,5
$INCLUDE CSF
$INCLUDE IO
$INCLUDE FIO
$INCLUDE WAIT
$FORTRAN
    SUBROUTINE LT(N)
    DIMENSION MES(10)
    DATA MFS/6HLISTAPZ
    MES(3)=N
    DO 1 J=4,10
1    MES(J)=0
    CALL SQ(MES)
    RETURN
    END
    SUBROUTINE LF(N)
    DIMENSION MES(10)
    DATA MFS/6HLFRAMEZ
    MES(3)=N
    DO 1 J=4,10
1    MES(J)=0
    CALL SQ(MES)
    RETURN
    END
    DIMENSION IFRAME(200)
    DIMENSION JDIR(1120),ITAPE(100),IOUT(100)
    DIMENSION MER1(24)
    DIMENSION MES(10)
    COMMON/SYSCOM/NDAT(100)
    DATA NT/0/
    DATA MFR1/30HNO FRAMES
    DATA NF/0/
    DATA MFS/6HFINFRMZ
    CALL WAIT(NDAT(94))
    NDAT(94)=-1
    CALL OPN(2)
    CALL RFW(2)
    CALL READW(2,1120;JDIR)
    NTAPE=JDIR(1)-1
    CALL IO(MES)
    DO 100 J=10,NTAPE/80
    CALL SCRA(2,10+J/8)
    CALL READW(2,1120;JDIR)
    DO 100 K=1,80
    L=1+(K-1)*14
    IF(JDIR(L+1).NE.MES(1))GO TO 100
    NT=NT+1
    ITAPE(NT)=JDIR(L)
    IOUT(NT)=0
100 CONTINUE
    CALL CLOS(2)
    NDAT(94)=0
    CALL WAIT(NDAT(93))
    NDAT(93)=-1

```

110

```

CALL OPN(3)
CALL REW(3)
CALL READW(3,1120;JDIR)
NFRAME=JDIR(1)-1
DO 200 J=10,NFRAME,80
CALL SCRA(3,10+J/8)
CALL READW(3,1120;JDIR)
DO 200 K=1,80
L=1+(K-1)*14
DO 201 I=1,NT
IF(ITAPE(I).EQ.JDIR(L+1))GO TO 202
201 CONTINUE
GO TO 200
202 NF=NF+1
IFRAME(NF)=JDIR(L)
IF(IOUT(I).EQ.0)CALL LT(ITAPE(I))
IOUT(I)=1
200 CONTINUE
CALL CLOS(3)
NDAT(93)=0
IF(NF.EQ.0)GO TO 400
DO 300 J=1,NF
300 CALL LF(IFRAME(J))
CALL EXIT
400 CALL TO(MER1)
CALL EXIT
END$

$CATALOG
TYPE=FG
NAME=FINFRM
ASSIGN 2=REELS,3=FRAMES
BEGIN
$EOJ.

```

///

\$JOB SAVEDK
\$ASSIGN,6,0
\$REW,5
\$INCLUDE CSF
\$INCLUDE FIO
\$INCLUDE IO
\$FORTRAN

DIMENSION J(2240)
DIMENSION MES(8)
DATA MFS/6HSAVEDK/
CALL IO(MES)
CALL OPN(2)
CALL OPN(3)
CALL OPN(4)
CALL RFW(2)
CALL RFW(3)
CALL RFW(4)
CALL READW(3,112,0)
CALL CLOS(3)
CALL WRITW(2,112,0)
DO 1 N=1,400
CALL READW(4,2240;J)
1 CALL WRITW(2,2240;J)
CALL RFW(2)
CALL RFW(4)
CALL CLOS(2)
CALL CLOS(4)
CALL EXIT
END\$

\$CATALOG
TYPE=FG
NAME=SAVEDK
\$ASSIGN 2=11,3=DKDIR,4=DK
BEGIN
\$EOJ

112

```

$JOB RESTDK
$ASSIGN,6,0
$REW,5
$INCLUDE FIO
$INCLUDE CSF
$INCLUDE IO
$FORTRA
    DIMENSION J(2240)
    DIMENSION MES(8),LK(10)
    DATA LK/6HLISDIR/
    DATA MES/6HRESTDK/
    CALL OPN(2)
    CALL RFW(2)
    CALL OPN(3)
    CALL RFW(3)
    CALL OPN(4)
    CALL RFW(4)
    CALL IO(MES)
    CALL READW(2,112,J)
    CALL WRITW(3,112,J)
    CALL RFW(3)
    CALL CLOS(3)
    CALL SQ(LK)
    DO 1 N=1,400
    CALL READW(2,2240;J)
1    CALL WRITW(4,2240;J)
    CALL RFW(2)
    CALL RFW(4)
    CALL CLOS(2)
    CALL CLOS(4)
    CALL EXIT
    ENDS
$CATALOG
TYPE=FG
NAME=RESTDK
ASSIGN 2=11,3=DKDIR,4=DK
BEGIN
$EOJ

```

113

\$JOB DELMRK
\$ASSIGN 6 0
\$REW,5
\$INCLUDE WAIT
\$FORTRAN

DIMENSION MES(8),JDIR(560)
DIMENSION MER(24)
COMMON/SYSCOM/NDAT(100)
DATA MFR/30HUNAUTHORIZED EI ***
DATA MES/6HDELMRK/
CALL IO(MES)
IF(MES(1).EQ.0)CALL ABORT
CALL WAIT(NDAT(92))
NDAT(92)=-1
CALL OPN(2)
IKEY=NDAT(95).AND.'40000000'
IF(IKEY.NE.0)GO TO 6999
CALL TO(MER)
CALL CLOS(2)
NDAT(92)=0
CALL EXIT

6999 MSEC=5*MOD(MES(1),100)
CALL SCRA(2,MSEC)
CALL READW(2,560,JDIR)
DO 1 J=1,560,7
IF(JDIR(J).NE.MES(1))GOTO 1
IF(JDIR(J+1).NE.MES(2))GOTO 1
KIND=JDIR(J+6)/4096
IF(MES(3).NE.KIND)GOTO 1
JDIR(J)=0
1 CONTINUE
CALL SCRA(2,MSEC)
CALL WRITW(2,560,JDIR)
CALL CLOS(2)
NDAT(92)=0
CALL EXIT
END\$

\$INCLUDE CSF
\$INCLUDE IO
\$INCLUDE FIO
\$CATALOG
TYPE=FG
NAME=DELMRK
ASSIGN 2=LANDMA
BEGIN
\$EOJ

114


```

$JOB WCELL
$ASSIGN,6,0
$REW,5
$INCLUDE CSF
$INCLUDE FIO
$INCLUDE IO
$FORTRAN
    DIMENSION J(560),MES(8),MOUT(24)
    COMMON/SYSCOM/NDAT(100)
    DATA MES/6HWCELL /
    CALL IQ(MES)
    IF(NDAT(92).LT.0)CALL WAIT(NDAT(92))
    NDAT(92)=-1
    CALL OPN(2)
    CALL RFW(2)
    MB=0
    MN=0
    DO 1 M=1,100
    CALL READW(2,560,0)
    MM=0
    DO 2 LI=1,560,7
    IF(J(LI).EQ.0)GOTO 2
    MM=MM+1
2    CONTINUE
    IF(MM.LT.MB)GOTO 1
    MB=MM
    MN=M-1
1    CONTINUE
    CALL CLOS(2)
    NDAT(92)=0
    ENCODE(72,333,MOUT)MN,MB
333  FORMAT('CELL',13,' HAS',14,' ENTRIES')
    CALL TO(MOUT)
    CALL EXIT
    ENDS
$INCLUDE WAIT
$CATALOG
TYPE=FG
NAME=WCELL
ASSIGN 2=LANDMA
BEGIN
$EOJ

```

115

```

$JOB,EDAY
$ASSIGN,6,0
$REW,5
$INCLUDE FIO
$INCLUDE CSF
$INCLUDE WAIT
$INCLUDE IO
$FORTRAN
    DIMENSION MES(8),J(560),MER(24)
    COMMON/SYSCOM/NDAT(100)
    DATA MFS/6HEDAY /
    DATA MFR/30HPRIVILEGED FUNCTION REFUSED /
    CALL IQ(MES)
    IF(NDAT(95).LT.0)GO TO 100
    CALL TO(MER)
    CALL EXIT
100  IF(NDAT(92).LT.0)CALL WAIT(NDAT(92))
    NDAT(92)=-1
    CALL OPN(2)
    N=MES(1)
    NSEC=5*MOD(N,100)
    CALL SCRA(2,NSEC)
    CALL READW(2,560,J)
    DO 200 L=1,560,7
    IF(J(L).EQ.N)J(L)=0
200  CONTINUE
    CALL SCRA(2,NSEC)
    CALL WRITW(2,560,J)
    CALL CLOS(2)
    NDAT(92)=0
    CALL EXIT
END$
$CATALOG
TYPE=FG
NAME=EDAY
$SSIGN 2=LANDMA
BEGIN
$EOJ

```

116

```

$JOB LISLMK
$ASSIGN 6 0
$REW,5
$INCLUDE IO
$INCLUDE CSF
$INCLUDE FIO
$INCLUDE WAIT
$FORTRA

```

```

SUBROUTINE LISTEM(JDIR,ID,KWANT)

```

```

  DIMENSION JDIR(560)

```

```

  DIMENSION LABEL(20)

```

```

  DIMENSION MOUT(24)

```

```

  DIMENSION INUM(20)

```

```

  DATA LABEL/48HL  L1 L2 L3 L4 L5 L6 L7 L8 L9      S 0      B E /

```

```

  DATA INUM/0,0,0,0,0,0,0,0,0,0,0,0,1,4,3,3,0,0,0,0,0/

```

```

  I1=0

```

```

  I2=0

```

```

  NL=0

```

```

  IL=LABEL(KWANT+1)

```

```

  NUMB=INUM(KWANT+1)

```

```

  IF(KWANT.EQ.12)GOTO 4444

```

```

  DO 4 JA=1,560,7

```

```

  KIND=JDIR(JA+6)/4096

```

```

  IF(KIND.GT.KWANT)GOTO 4

```

```

  IF(KIND.LT.KWANT.AND.KWANT.NE.9)GOTO 4

```

```

  IF(JDIR(JA).NE.ID)GOTO 4

```

```

  IL=LABEL(KIND+1)

```

```

  J=JA+1

```

```

  NL=NL+1

```

```

  IT=JDIR(J)

```

```

  J1=JDIR(J+1)

```

```

  J2=JDIR(J+2)

```

```

  K1=JDIR(J+3)

```

```

  K2=JDIR(J+4)

```

```

  L1=JDIR(J+5)

```

```

  L1=MOD(L1,4096)

```

```

  KF=0

```

```

  IF(I1.EQ.J1.AND.I2.EQ.J2)KF=1

```

```

  IF(J1.EQ.0.AND.J2.EQ.0)KF=1

```

```

333  FORMAT(A3,I8,I3,4I8)

```

```

334  FORMAT(A3,8I8)

```

```

  ENCODE(72,333,MOUT)IL,IT,L1,K1,K2,J1,J2

```

```

  IF(KF.EQ.1)ENCODE(72,333,MOUT)IL,IT,L1,K1,K2

```

```

  KA=JA+2

```

```

  KB=KA+NUMB-1

```

```

  IF(IT.NE.0)KA=KA-1

```

```

  IF(NUMB.NE.0)ENCODE(72,334,MOUT)IL,(JDIR(KL),KL=KA,KB)

```

```

68  CONTINUE

```

```

  I1=J1

```

```

  I2=J2

```

```

  CALL TO(MOUT)

```

```

4  CONTINUE

```

```

  RETURN

```

```

4444 IH=0

```

```

  DO 44 JA=1,560,7

```

```

  IF(JDIR(JA).NE.ID)GOTO 44

```

```

  KIND=JDIR(JA+6)/4096

```

117

```

      IF(KIND.NE.12)GOTO 45
      IH=IH+1
      I1=JDIR(JA+2)
      I2=JDIR(JA+3)
      I3=JDIR(JA+4)
      I4=JDIR(JA+5)
      GOTO 44
45    IF(KIND.NE.13)GOTO 44
      IH=IH+1
      I5=JDIR(JA+2)
      I6=JDIR(JA+3)
      I7=JDIR(JA+4)
44    CONTINUE
      ENCODE(72,339,MOUT)I1,I2,I3,I4,I5,I6,I7
      IF(IH.EQ.2)CALL TQ(MOUT)
339   FORMAT('0',8I8)
      RETURN
END
-----
      DIMENSION JDIR(560),MES(8),MOUT(24),MER(24)
      DIMENSION NAM(2)
      COMMON/SYSCOM/NDAT(100)
      DATA NAM/6HLISLMKZ
      DATA MER/30H NO LANDMARKS ON SPECIFIED DAY
199   MES(1)=NAM(1)
      MES(2)=NAM(2)
      CALL IO(MES)
      IF(MES(1).EQ.NAM(1))GO TO 299
      IF(MES(1).EQ.0)CALL ABORT
      IF(NDAT(92).LT.0)CALL WAIT(NDAT(92))
      NDAT(92)=-1
      CALL OPN(2)
      MSEC=5*MOD(MES(1),100)
      CALL SCRA(2,MSEC)
      CALL READW(2,560,JDIR)
      CALL CLOS(2)
      NDAT(92)=0
      JUMP=MES(2)
      IF(JUMP.NE.0)GOTO 4444
      DO 450 JA=1,4
      JB=13-JA
450   CALL LISTEM(JDIR,MES(1),JB)
      GOTO 199
4444  IF(JUMP.LT.1)CALL EXIT
      IF(JUMP.GT.5)CALL EXIT
      GOTO(203,202,201,204,205),JUMP
C LIST LANDMARKS
201   CONTINUE
      CALL LISTEM(JDIR,MES(1),9)
      GO TO 199
299   CONTINUE
      CALL EXIT
C LIST SPIN RATE
202   CONTINUE
      CALL LISTEM(JDIR,MES(1),11)
      GO TO 199
C LIST ORBIT

```

118

```
203 CONTINUE
    CALL LISTEM(JDIR,MES(1),12)
    GOTO 199
C BETA-BETADOT
204 CONTINUE
    CALL LISTEM(JDIR,MES(1),14)
    GOTO 199
C EARTH EDGES
205 CONTINUE
    CALL LISTEM(JDIR,MES(1),15)
    GOTO 199
END$
$CATALOG
TYPE=FG
NAME=LISLMK
ASSIGN 2=LANDMA
BEGIN
$EOJ
```

119

```

$JOB LISDIR
$ASSIGN 6 0
$REW,5
$INCLUDE CSF
$INCLUDE IO
$INCLUDE FIO
$FORTRA
    DIMENSION MES(8),J(112)
    DIMENSION MOUT(24)
    COMMON/SYSCOM/NDAT(100)
    DATA MES/6HLISDIR/
    CALL IO(MES)
    CALL OPN(2)
    CALL RFW(2)
    CALL RFADW(2,112,J)
    CALL CLOS(2)
1   FORMAT(I3,5I7,3I3,I9)
    DO 11 K=1,8
    L=14*(K-1)+1
    M=L+6
    MK=NDAT(K+82)
    MJ=J(L+7)
    IF(J(L).EQ.0)GOTO 11
    ENCODE(72,1,MOUT)K,MJ,(J(N),N=L,M),MK
    CALL TO(MOUT)
11  CONTINUE
    CALL EXIT
    ENDS
$CATALOG
TYPE=FG
A =
VSWIGPIS=LISDIR
BEGIN
$EOJ

```

```

$JOB LFRAME
$ASSIGN,6,0
$REW,5
$INCLUDE CSF
$INCLUDE FIO
$INCLUDE FTIME
$INCLUDE GETGAM
$INCLUDE ILALO
$INCLUDE IO
$INCLUDE LOOKUP
$INCLUDE SATFAR
$INCLUDE TVSAT
$INCLUDE WAIT
$FORTRAN
    DIMENSION MES(8),NAM(2)
    DATA NAM/6HLFRAME/
1    MES(1)=NAM(1)
    MES(2)=NAM(2)
    CALL IO(MES)
    IF(MES(1).EQ.NAM(1).AND.MES(2).EQ.NAM(2))CALL EXIT
    DO 2 I=1,8
    IF(MES(I).NE.0)CALL DOIT(MES(I))
2    CONTINUE
    GO TO 1
    END
    SUBROUTINE DOIT(N)
    COMMON/SYSCOM/ICOM(100)
    EQUIVALENCE (ICOM(2),IDAY)
    DIMENSION M(14),MOUT(24)
    CALL LOOK(3,N,M)
    ITL=M(5)
    ITE=M(6)
    CALL TVSAT(N,ITL,ITE,IL2,IF2,IT,ID)
    IF(ID.EQ.0)CALL EXIT
    NTAPE=M(2)
    NLIN=M(7)
    ENCODE(72,100,MOUT)N,NTAPE,ID,IT,IL2,IE2,NLIN
100  FORMAT('F',I5,2I7,I8,3I7,2I9)
    IF(IDAY.NE.ID)GO TO 1
    INAV=1
    PICTIM=FTIME(IT)
    XLIN=IL2
    XELE=IF2
    CALL GETGAM(ID,IT,BETA IN,BETDOT)
    CALL SATFAR(PICTIM,XLIN,XELE,XLAT,XLON,1,INAV,BETA IN,BETDOT,0.0)
    ILAT=IIALO(XLAT)
    ILON=IIALO(XLON)
    ENCODE(72,100,MOUT)N,NTAPE,ID,IT,IL2,IE2,NLIN,ILAT,ILON
1    CALL TO(MOUT)
    RETURN
    ENDS
$CATALOG
TYPE=FG
NAME=LFRAME
ASSIGN 2=REELS,3=FRAMES,11=LANDMA
BEGIN
$EOJ

```

```

$JOB LNDMRK
$ASSIGN,6,0
$REW,5
$INCLUDE TVSAT
$INCLUDE WAIT
$INCLUDE LOOKUP
$INCLUDE CSF
$INCLUDE FIO
$INCLUDE IO
$FORTRAN
    DIMENSION MIN(8),MOUT(10),NAM(2)
    DATA NAM/6HLNDRMK/
    DATA MOUT/6HDLANDM/
C INPUT IS FRAME, LAT, LON, TVL, TVE, KEY
C OUTPUT IS SYYDDD, HHMMSS, KEY, SATL, SATE, LAT, LON
1   MIN(1)=NAM(1)
    MIN(2)=NAM(2)
    CALL IO(MIN)
    IF(MIN(1).EQ.NAM(1))CALL EXIT
C
    CALL TVSAT(MIN(1),MIN(4),MIN(5),IL,IE,IT,ID)
C ID=0 IMP MISSING FRAME OR TAPE LINKAGE
    IF(ID.EQ.0)CALL EXIT
C SYYDDD
    MOUT(3)=ID
C HHMMSS
    MOUT(4)=IT
C KEY
    MOUT(5)=MIN(6)
C SATLIN
    MOUT(6)=IL
C SAT ELEMENT
    MOUT(7)=IE
C LAT
    MOUT(8)=MIN(2)
C LON
    MOUT(9)=MIN(3)
C KIND
    MOUT(10)=MIN(7)
    CALL SQ(MOUT)
    GOTO 1
END$
$CATALOG
TYPE=FG
NAME=LNDMRK
ASSIGN 2,REELS,3=FRAMES
BEGIN
$EOJ

```

122


```

$JOB INIT
$REW,5
$INCLUDE IO
$INCLUDE FIO
$ASSIGN 6 0
$FORTRAN
    DIMENSION JSEC(112)
    COMMON/SYSCOM/NDAT(100)
101  FORMAT(' TYPE IN 4 DIGIT TIME, HOURS AND MINUTES')
102  FORMAT(2I2)
901  FORMAT('//1X,8A3//')
    CALL OPN(2)
    CALL SCRA(2,0)
    CALL READW(2,112,JSEC)
    CALL CLOS(2)
    CALL FROGS(1,5HOPCOM,7)
    WRITE(1,901)(JSEC(JJ),JJ=1,8)
    WRITE(1,101)
    READ(1,102)I1,I2
    DO 99 JJ=1,100
99   NDAT(JJ)=0
    I=60*I2+3600*I1
    NDAT(96)=I
    CALL EXIT
    ENDS
$CATALOG
TYPE=FG
NAME=INIT,2
$ASSIGN 1=1,2=VERS,3=SYSCOM
BEGIN
$EOJ

```

123

\$JOB ADDR
\$ASSIGN 6 0

\$Z\$W,GN 5 LR
\$FORTRA

SUBROUTINE ADDR(ID,IT,IL,IE,IDL,IDE,IS,IM)

C ***** INPUTS *****
C ID IS DAY NUMBER
C IL IS SATELLITE START LINE
C IE IS SATELLITE START ELEMENT
C ***** OUTPUTS *****
C IL IS CHANGED TO LIE ON A LINE IN QUE
C IE IS CHANGED TO LIE ON WORD BOUNDARY
C IDL IS LINE SAMPLING ON DISK
C IDE IS ELEMENT SAMPLING ON DISK
C IS IS STARTING SECTOR FOR REQUEST
C IF IS = -1 IMP NON-HIT
C IM IS WORD OFFSET FROM STARTING SECTOR

DIMENSION J(112)

DATA N/0/

IS=-1

IF(N.EQ.0)CALL OPN(7)

N=1

CALL RFW(7)

CALL READW(7,112,J)

DO 1 K=1,8

L=(K-1)*14+1

IF(ID.NE.J(L))GO TO 1

L=L+1

IF(IT.NE.J(L))GOTO 1

L=L+1

ILS=J(L)

L=L+1

IES=J(L)

L=L+1

LD=J(L)

L=L+1

IED=J(L)

IF(IL.LT.ILS)GOTO 1

IF(IE.LT.IES)GOTO 1

IF(IL.GT.ILS+LD*500)GOTO 1

IF(IE.GT.IES+IED*224*3)GOTO 1

* * * (IL,IE) NOW LIES WITHIN AREA OF INTEREST * * *

INC=3*IED

M=(IL-ILS)/LD

IL=ILS+M*LD

IS=2*M

IM=(IE-IES)/INC

IE=IES+IM*INC

IDL=LD

IDE=IED

IS=IS+1000*(K-1)

CALL AGE(K)

IF(IM.IT.112)GO TO 1

IM=IM-112

IS=IS+1

124

```
1    CONTINUE
C IM SHOULD COUNT FROM 1
  IF (IS.NE.-1) IM=IM+1
  RETURN
END$

$FILEMA
ESTAB,5,ADDR,0,1,0,4
EXIT
$EOJ
```

125

```

$ASSIGN,6,0
$REW,5
$INCLUDE CSF
$INCLUDE HIO
$INCLUDE IK
$INCLUDE IO
$INCLUDE FIO
$FORTRA
    COMMON/SYSCOM/NDAT(100)
    DIMENSION MES(8),M(64)
    DATA MFS/6HENH001/
    CALL IO(MES)
    CALL OPN(2)
    CALL RFW(2)
    CALL READW(2,64,M)
    CALL RFW(2)
    IF(MES(1),NE.0)GOTO 100
    DO 1 J=1,64
1      M(J)=0
100    D=MES(2)-MES(1)
        IF(D.LT.1.)D=1.
        S1=MES(4)-MES(3)
        S2=MES(6)-MES(5)
        S3=MES(8)-MES(7)
        S1=S1/D
        S2=S2/D
        S3=S3/D
        X1=MES(3)
        X2=MES(5)
        X3=MES(7)
        I=MES(1)
        J=MES(2)
        DO 200 L=1,J
            M1=X1
            M2=X2
            M3=X3
            X1=X1+S1
            X2=X2+S2
            X3=X3+S3
200    M(L)=4096*M1+64*M2+M3
        CALL WRITW(2,64,M)
        CALL CLOS(2)
        IF(J.NE.63)CALL EXIT
        CALL HIO('101','2140','1000000)
        CALL HIO('101','2000','23000000+IK(M(1)))
        DO 300 L=2,64
300    CALL HIO('101','2000','21000000+IK(M(L)))
        CALL HIO('101','2140',NDAT(75))
        CALL EXIT
    ENDS
CATALOG
TYPE=FG
NAME=ENH001
ASSIGN 2=ENH1
BEGIN
FILEMA
CREATE,ENH1,0,1,3,1
EXIT
EDJ

```

126

```

$JOB LOGOUT
$ASSIGN 6 0
$REW,5
$INCLUDE CSF
$INCLUDE FIO
$INCLUDE IO
$FORTRAN

```

```

COMMON/SYSCOM/NCOM(100)
DIMENSION MCOM(100)
DIMENSION MES(8),MOUT(24),JLOG(112)
DIMENSION MMOUT(12)
DIMENSION MESX9(10)
EQUIVALENCE(MOUT(12),MMOUT(1))
DATA MFS/6HLOGOUT/
DATA MFSX9/6HDOI0XX/
  CALL FROGS(7,MES)
  CALL IO(MES)
  IF(NCOM(1).EQ.0)CALL EXIT
  KPROJ=NCOM(1)
  NCOM(1)=0
  DO 983 JA=1,100
983  MCOM(JA)=NCOM(JA)
  CALL OPN(3)
  CALL RFW(3)
  CALL WRITW(3,100,MCOM)
  CALL CLOS(3)
  NCOM(76)=1
  CALL SQ(MESX9)
  NCLOCK=NCOM(97)
  CALL OPN(2)
  CALL RFW(2)
  IF(MES(1).EQ.987898)GO TO 900
  CALL READW(2,112,JLOG)
3  CONTINUE
  CALL SCRA(2,0)
  DO 2 J=1,112,2
  IF(JLOG(J).EQ.0.OR.JLOG(J).EQ.KPROJ)GO TO 100
2  CONTINUE
  J=111
100 JLOG(J)=KPROJ
  JLOG(J+1)=JLOG(J+1)+NCLOCK
  CALL WRITW(2,112,JLOG)
  CALL CLOS(2)
  F60=NCLOCK
  F60=F60/60.
  ENCODE(72,222,MOUT)KPROJ,F60
222 FORMAT('PROJECT',I5,F8.2,' MINUTES')
  NCOM(96)=NCOM(96)+NCOM(97)
  IF(NCOM(96).GT.86400)NCOM(96)=NCOM(96)-86400
  NCOM(97)=0
  MM=NCOM(96)/60
  MM=MOD(MM,60)
  M1=MOD(MM,10)
  M2=MOD(MM/10,10)
  MM=NCOM(96)/3600
  M3=MOD(MM,10)
  M4=MM/10

```

127

```
ENCODE(36,333,MMOUT,M4,M3,M2,M1
333  FORMAT('SYSTEM IDLE AT ',4I1)
      CALL TO(MOUT)
      CALL EXIT
900  DO 901 J=1,112
901  JLOG(J)=0
      GO TO 3
END$
```

```
$CATALOG
TYPE=FG
NAME=LOGOUT
ASSIGN 2=LOG,3=SYSCOM
BEGIN
$EOJ
```

```

$JOB LDCNTR
$ASSIGN 6,0
$REW,5
$INCLUDE CSF
$INCLUDE IO
$INCLUDE FIO
$INCLUDE LOOKUP
$INCLUDE WAIT
$INCLUDE AGE
$INCLUDE STAGE
$INCLUDE NRZL
$INCLUDE FTIME
$FORTRA
  SUBROUTINE POS(M,LINE,ITIME )
    DIMENSION DUMMY(2)
    COMMON/SYSCOM/ICOM(100)
    DIMENSION MOUT(24)
    DATA MOUT/40HTAPE IDENTITY VERIFIED BY LDCNTR /
    DATA KEY/0/
    CALL OPN(2)
    CALL RFW(2)
    N=M-1
    IF(N.LT.1)RETURN
    T=FTIME(ITIME)
    DO 1 J=1,N
    IF(ICOM(1).EQ.0)GOTO 222
    IF(KEY.EQ.0)GOTO 100
    CALL READW(2,2,DUMMY)
    GOTO 1
100  CALL NRZL(L,IT)
    IF(L.EQ.0)GOTO 1
    I=L-J
    IF(IABS(I-LINE).GT.5)GOTO 1
    X=FTIME(IT)
    O=.6*FLOAT(L)/3600.
    X=X-O
    Y=ABS(X-T)*3600.
    IF(Y.GT.100.)GOTO 1
    KEY=1
    CALL TO(MOUT)
1  CONTINUE
    CALL CLOS(2)
    RETURN
222  CALL RFW(2)
    CALL CLOS(2)
    CALL EXIT
    END
  SUBROUTINE PICK(N)
    DIMENSION NMES(24)
    COMMON/SYSCOM/NDAT(100)
    DATA NMES/30HAREA 0 USED BY LDCNTR /
    I=1
    IO=0
    DO 1 J=1,8
    IF(IO.GT.NDAT(J+82))GO TO 1

```

129

```

      I=J
      IO=NDAT(J+82)
1     CONTINUE
      CALL AGE(I)
      NMES(2)=NMES(2)+I
      CALL TO(NMES)
      N=I
      RETURN
      END
      DIMENSION MES(8),M1(14),MOUT(10),MER(24),ISEC(112)
      COMMON/SYSCOM/NDAT(100)
      DATA MFS/6HLD CNTR/
      DATA MFR/30HLD CNTR PARAMETER ERROR
      DATA MOUT/6HABSLD /
      CALL IO(MES)
      CALL WAIT(NDAT(94))
      NDAT(94)=-1
      CALL LOOKUP(4,MES(1),M1)
      NDAT(94)=0
      IF(M1(1).EQ.0)GO TO 900
      LS=MES(2)-M1(4)+1
      IF(LS.LT.1)GO TO 900
      IES=MES(3)-M1(5)+1
      IF(IES.LE.0)GO TO 900
C FIRST TAPE RELATIVE ELEMENT TO LOAD
      MOUT(4)=IES
C LINE INCREMENT
      IF(MES(4).EQ.0)MES(4)=1
      MOUT(5)=MES(4)
C ELEMENT INCREMENT
      IF(MES(5).EQ.0)MES(5)=3
      MOUT(6)=MES(5)
C SATELLITE LINE
      MOUT(7)=MES(2)
C FIRST SATELLITE LMENT ON TAPE
      MOUT(8)=M1(5)
C EEC KEY
      MOUT(9)=MES(6)
C TAPE NUMBER
      MOUT(10)=M1(1)
      CALL PICK(NSEC)
C AREA TO PUT DATA
      MOUT(3)=NSEC
      CALL POS(LS,M1(4),M1(3))
      CALL OPN(3)
      CALL RFW(3)
      CALL RFADW(3,112,ISEC)
      M=14*(NSEC-1)+1
C $YYDDD
      ISEC(M)=M1(2)
      M=M+1
C HHMMSS
      ISEC(M)=M1(3)

```

130


```

      M=M+1
      DO 99 J=2,6
        ISEC(M)=MES(J)
99    M=M+1
C TAPENO
      ISEC(M)=M1(1)
      CALL RFW(3)
      CALL WRITW(3,112,ISEC)
      CALL CLOS(3)
      CALL SQ(MOUT)
      CALL EXIT
900  CALL TO(MER)
      CALL EXIT
      ENDS
$CATALOG
TYPE=FG,PRIV
NAME=LDCNTR
ASSIGN 2=11,3=DKDIR,4=REELS
BEGIN
$EOJ

```

```

$JOB ABSLD
$ASSIGN,6,0
$REW,5
$INCLUDE CSF
$INCLUDE EDGFCO
$INCLUDE FIO
$INCLUDE FTIME
$INCLUDE GETGAM
$INCLUDE IO
$INCLUDE LOOKUP
$INCLUDE SATFAR
$INCLUDE STAGE
$INCLUDE WAIT
$FORTRAN
    DIMENSION IS(8193)
    DIMENSION IT(2731)
    DIMENSION ID(1120)
    DIMENSION MES(8)
    DIMENSION MUNG(14)
    DIMENSION MOUT(24)
    COMMON/SYSCOM/ICOM(100)
    EQUIVALENCE(IT(1),IS(5462))
    DATA MES/6HABSLD /
    DATA MOUT/30HTAPE LOAD COMPLETED
    CALL OPN(2)
    CALL OPN(3)
    CALL IO(MES)
C MES(1)= DISK AREA 1 - 8
    NSEC=MES(1)
C MES(2)=STARTING ELEMENT TAPE RELATIVE
    IES=MES(2)
C MES(3)IS LINE INCREMENT
    ILD=MES(3)
    IF(ILD.EQ.0)ILD=1
C MES(4) IS ELEMENT INCREMENT
    IED=MES(4)
    IF(IED.EQ.0)IED=3
C MES(5) IS SATELLITE LINE TAPE IS NOW POSITIONED AT
    IPL=MES(5)
C MES(6) IS FIRST SAT ELEMENT ON TAPE
    IOF=MES(6)
C MES(7) IS REC KEY
    KEE=MES(7)
C MES(8) IS REFL NUMBER
    IF(ICOM(94).LT.0)CALL WAIT(ICOM(94))
    ICDM(94)=-1
    CALL LOOKUP(4,MES(8),MUNG)
    ICOM(94)=0
    IDAY=MUNG(2)
    PST=FTIME(MUNG(3))
    CALL GETGAM(IDAY,MUNG(3),BFTA,BDOT
    NSEC=(NSFC-1)*1000
    NR=500*ILD
    NE=672*IFD
    IEL=IES+NE-1

```

132

```

      INAV=1
      DO 1 J=1,500
      IF(ICOM(1).EQ.0)GOTO 222
C READ TAPE RECORD
      CALL READW(2,2731,IT)
C MOVE SAMPLES SPLIT TO IS
C FIRST TAPE WORD TO START SPLITTING
      ITST=1+(IES-1)/3
      NSPLIT=NF+6
C OFFSET BETWEEN SPLIT START AND FIRST WANTED ELEMENT
      IR=MOD(IFS+2,3)
C PLACE IN SATELLITE COORDINATE SYSTEM TO PUT SPLIT SAMPLES
      IDEST=IOF+IES-IR -1
      IF(KEE.EQ.0)CALL CRACK(NSPLIT,IT(ITST),IS(IDEST))
C IF EEC NEEDED, MUST SPLIT WHOLE LINE
      IF(KEE.NF.0)CALL CRACK(8193-IOF,IT,IS(IOF))
C DO EARTH-EDGE CORRECTION
      CALL EDGFCO(PST,KEE,INAV,BFTA,BDOT,IPL,IS,IBELT,IBAD)
      IF(IBAD.FQ.0)GOTO 777
      DO 776 J7=1,8193
776  IS(J7)=0
777  CONTINUE
      IPL=IPI+ILD
C K IS PLACE IN OUTPUT BUFFER TO PUT LINE
      K=MOD(J,5)
      IF(K.EQ.0)K=5
      K=K-1
      KK=0
      JOF=IOF+IDELT
      DO 2 JJ=IES,IEL,IED
      M=0
      JA=JOF-1+JJ
      DO 22 JJJ=1,IED
      MM=JA+JJJ-1
22  M=M+IS(MM)
      M=M/IED
      KK=KK+1
2  IS(KK)=M
      KAD=224*K+1
      CALL PACK(672,IS,ID(KAD))
      IF(K.NF.4)GO TO 11
      CALL SCRA(3,NSEC)
      CALL WRITW(3,1120,ID)
      NSEC=NSEC+10
11  IF(ILD.EQ.1)GOTO 1
      II=ILD-1
      DO 12 IX=1,II
12  CALL READW(2,2,IT)
1  CONTINUE
      CALL TQ(MOUT)
222  CALL RFW(2)
      CALL CLOS(2)
      CALL CLOS(3)
      CALL EXIT
      ENDS

```

133

\$CATALOG
NAME=ABSLD
TYPE=FG
TBSIZE=5000
ASSIGN 2=11,3=DK,4=REELS,11=LANDMA
BEGIN
\$EOJ

134

\$JOB LOGGIN
\$ASSIGN 6 0
\$REW,5
\$FORTRAN

```
COMMON/SYSCOM/N(100)  
DIMENSION MESX9(10)  
DIMENSION MES(8)  
DIMENSION MME1(2),MME2(10)  
DIMENSION MES2(24),MFS3(24),MES4(24),MES5(24),MOUT(24)  
DIMENSION MC(100)  
DIMENSION MPROJ(112)  
DIMENSION MMOUT(13)  
EQUIVALENCE(MOUT(11),MMOUT(1))  
DATA MC/100*0/  
DATA MFSX9/6HDDIOXX/  
DATA MFS5/30H***PRIVILEGED PROJECT***  
DATA IRLK/3H /  
DATA MFS/6HLOGGIN/  
DATA MFS4/30HILLEGAL PROJECT  
DATA MFS2/30HMUST LOGOUT FIRST  
DATA MFS3/6HLOGOUT/  
DATA MME1/6HENH001/  
DATA MME2/0,0,0,63,0,63,0,63,0,63/  
MES3(3)=0
```

```
MME2(1)=MME1(1)  
MME2(2)=MME1(2)  
CALL OPN(4)  
CALL RFW(4)  
CALL READW(4,90,MC)  
CALL CLOS(4)
```

C COMMON 1-90 FROM DISK
C COMMON 91-94 ZERO
C COMMON 95-100 UNCHANGED

DO 983 JA=1,95

```
983 N(JA)=MC(JA)  
CALL IO(MES)  
IF(MES(1).EQ.0.OR.MES(1).GT.9999)CALL EXIT  
IF(N(1).EQ.0)GO TO 111  
CALL TO(MES2)  
CALL EXIT
```

```
111 CONTINUE  
N(80)=0  
N(79)=177  
N(78)=1400  
N(76)=2  
N(75)=100410000  
N(74)=0  
N(73)=0  
N(72)=102020202  
CALL SQ(MESX9)  
CALL SQ(MME2)  
NTICK=MES(2)  
IF(NTICK.EQ.0)NTICK=10  
NTICK=NTICK*60  
NTICK=NTICK*120  
CALL FROGS(6,6HLOGOUT,254,NTICK)
```

135

```

CALL OPN(3)
CALL RFW(3)
CALL READW(3,10,MOUT)
CALL CLOS(3)
N(1)=MFS(1)
N(96)=N(96)+N(97)
N(97)=0
IF(N(96).GT.86400)N(96)=N(96)-86400
MM=N(96)/60
MM=MOD(MM,60)
M1=MOD(MM,10)
M2=MOD(MM/10,10)
MM=N(96)/3600
M3=MOD(MM,10)
M4=MM/10
ENCODE(42,222,MMOUT,M4,M3,M2,M1
222 FORMAT(' ACTIVE AT ',12,311)
CALL TO(MOUT)
CALL OPN(2)
CALL SCRA(2,0)
CALL READW(2,112,M PROJ)
CALL CLOS(2)
DO 93 JJ=1,112,2
IF(M PROJ(JJ).EQ.N(1))GO TO 993
93 CONTINUE
CALL TO(MES4)
CALL SQ(MES3)
CALL EXIT
993 N(95)=M PROJ(JJ+1)
IKEY=N(95).AND.'40000000
IF(IKEY.FQ.0)CALL EXIT
CALL TO(MES5)
CALL EXIT
END$

```

```

$INCLUDE IO
$INCLUDE FIO
$INCLUDE CSF
$CATALOG
TYPE=FG
NAME=LOGGIN
ASSIGN 2=PROJS,3=VERS,4=SYSCOM
BEGIN
$EDJ

```

136

```

$JOB DOIOXX
$REW,5
$ASSIGN,6,0
$INCLUDE CSF
$INCLUDE HIO
$FORTRA
    DIMENSION MES(8)
    COMMON/SYSCOM/N(100)
    DATA MFS/6HDOIOXX/
    CALL IO(MES)
    CALL HIO('101','2000*16,N(80))
    CALL HIO('101','2000*2*16,N(79))
    CALL HIO('101','2000*3*16,N(78))
    CALL HIO('101','2000*5*16,N(76))
    CALL HIO('101','2000*6*16,N(75))
    CALL HIO('101','1000*3*16,N(74))
    CALL HIO('101','1000*5*16,N(73))
    CALL HIO('101','1000*6*16,N(72))
    CALL EXIT
    ENDS
$CATALOG
TYPE=FG
NAME=DOIOXX
BEGIN
$EOJ

```

\$JOB CNTRL
\$REW,5
\$ASSIGN,6,0
\$ASSEMB

START BLL \$MAIN
BLU \$EXIT
ENDS START

\$FORTRAN

SUBROUTINE MAIN
DIMENSION MES(8),NAM(2),MOUT(24)
DIMENSION MSYS(10)
DIMENSION LMES(3)
DIMENSION LF(2)
DIMENSION NF(2)
DIMENSION IW(2)
DIMENSION LD(2)
DIMENSION JE(2)
DIMENSION LKMES(3)
DIMENSION MTVL(10)
DIMENSION MECHO(24)

C NDAT(55) IS CURSOR POINTER WORD
C NDAT(56) IS CURRENT-FRAME/DESIRED FRAME
C 57 IS FRAME ROUNDS
C 59 IS LAT
C 60 IS LONG
C 61 IS WIND TFMP STORAGE
C 62 IS LAST AREA USED FOR A LOAD
C 63
C 64 IS CURSOR SIZE

COMMON/SYSCOM/NDAT(100)
DATA MECHO/3H * /
DATA IW/6HBAKGND/
DATA NF/6HNAVFRM/
DATA LF/6HLFRAME/
DATA LD/6HLNDMRK/
DATA JF/6HFRMEAR/
DATA LMES/6HLOOP /
DATA LKMES/6HLISDIR/
DATA MTVL/6HLDCNTV/
DATA MOUT/30H BAD CNTRL
DATA NAM/6HCNTRL /

999 MES(1)=NAM(1)
MES(2)=NAM(2)
CALL IQ(MES)
K=MES(1)
IF(MES(1).EQ.NAM(1))RETURN
IF(NDAT(1).EQ.0)GOTO 999
MECHO(1)=MECHO(1).AND.'77777400
MECHO(1)=MECHO(1).OR.K

C L IF(K.NF.'314)GOTO 2
NDAT(62)=0
CALL SQ(LMES)

138


```

      GOTO 1
C   A
2   IF(K.NF.'301')GOTO 3
      NDAT(62)=0
      CALL STEP
      GOTO 1
C   B
3   IF(K.NF.'302')GOTO 4
      NDAT(62)=0
      CALL BACK
      GOTO 1
C   R
4   IF(K.NF.'322')GOTO 5
      NDAT(62)=0
      CALL RST
      GOTO 1
C   C
5   IF(K.NF.'303')GOTO 6
      MSYS(1)=IF(1)
      MSYS(2)=IF(2)
      MSYS(3)=NDAT(56).AND.'7777
      DO 500 JA=4,10
500  MSYS(JA)=0
      CALL SQ(MSYS)
      GOTO 1
C M
6   IF(K.NF.'315')GOTO 7
      NDAT(62)=0
      MSYS(1)=LD(1)
      MSYS(2)=LD(2)
      MSYS(3)=NDAT(56).AND.'7777
      MSYS(4)=NDAT(59)
      MSYS(5)=NDAT(60)
      MSYS(6)=NDAT(55)/4096
      MSYS(7)=MOD(NDAT(55),4096)
      MSYS(8)=0
      CALL SQ(MSYS)
      CALL STEP
      GOTO 1
C E
7   IF(K.NF.'305')GOTO 8
      MSYS(1)=JE(1)
      MSYS(2)=JE(2)
      MSYS(3)=NDAT(56).AND.'7777
      MSYS(4)=NDAT(55)/4096
      MSYS(5)=MOD(NDAT(55),4096)
      CALL SQ(MSYS)
      GOTO 1
C N
8   IF(K.NF.'316')GOTO 9
      CALL SQ(NF)
      GOTO 1
C W
9   IF(K.NF.'327')GOTO 10
      NDAT(62)=0
      IF(NDAT(61).NE.0)GOTO 900
      NDAT(61)=NDAT(55)

```

139

```

CALL STEP
GOTO 1
900 MSYS(1)=IW(1)
MSYS(2)=IW(2)
MSYS(6)=NDAT(56).AND.'7777
MSYS(3)=MSYS(6)-1
IT=NDAT(57)/4096
IF(MSYS(6).EQ.IT)CALL STEP
KK=NDAT(61)
NDAT(61)=0
MSYS(4)=KK/4096
MSYS(5)=MOD(KK,4096)
MSYS(7)=NDAT(55)/4096
MSYS(8)=MOD(NDAT(55),4096)
MSYS(9)=NDAT(64)/4096
MSYS(10)=MOD(NDAT(64),4096)
CALL SQ(MSYS)
GOTO 1

C B
10 IF(K.NF.'304)GOTO 11
CALL SQ(LKMES)
GOTO 1

C 1 THRU 8
11 IF(K.LT.'260)GOTO 12
IF(K.GT.'270)GOTO 12
MTVL(3)=K-'260
NDAT(62)=MTVL(3)
MTVL(4)=NDAT(56).AND.'7777
MTVL(5)=0
MTVL(6)=0
MTVL(7)=1
CALL SQ(MTVL)
GOTO 1

C P J K
12 IF(K.LT.'311.OR.K.GT.'313)GOTO 13
K=K-'307
MTVL(3)=NDAT(62)
MTVL(4)=NDAT(56).AND.'7777
MTVL(7)=K
NDAT(62)=0

C CURSOR
ICL=NDAT(55)/4096
ICE=MOD(NDAT(55),4096)
ISZ=500/K
ISZ=ISZ/2
MTVL(5)=ICL-ISZ
ISZ=700/K
ISZ=ISZ/2
MTVL(6)=ICE-ISZ
CALL SQ(MTVL)
GOTO 1
13 CONTINUE
GOTO 999
1 CALL TO(MECHO)
GOTO 999

```

140

```

END
SUBROUTINE STEP
COMMON/SYSCOM/NDAT(100)
IT=NDAT(57)/4096
IB=NDAT(57)-4096*IT
ID=MOD(NDAT(56),4096)
L=IT-IR
IF(ID.FQ.IT)GOTO 2
NDAT(56)=NDAT(56)+1
RETURN
2 NDAT(56)=NDAT(56)-L
RETURN
END
SUBROUTINE BACK
COMMON/SYSCOM/NDAT(100)
IT=NDAT(57)/4096
IB=NDAT(57).AND.'7777'
ID=NDAT(56).AND.'7777'
L=IT-IR
IF(ID.FQ.IB)GOTO 2
NDAT(56)=NDAT(56)-1
RETURN
2 NDAT(56)=NDAT(56)+L
RETURN
END$
$INCLUDE CSF
$CATALOG
TYPE=FG
NAME=CNTRL
BEGIN
$EOJ

```

141

```

$JOB CC
$REW,5
$ASSIGN 6,0
$ASSEMB
START    BLL    $MAIN
          BLU    $EXI
          ENDS   START
$INCLUDE,CSF
$INCLUDE,STAGE
$INCLUDE,IO
$FORTRAN
  SUBROUTINE MAIN
  DIMENSION LIS(300)
  DIMENSION MES(27);M(60)
  DIMENSION NERN(2)
  DIMENSION NER(2)
  DIMENSION M3(24)
  DIMENSION NAM(2)
  COMMON/SYSCOM/IP(100)
  DATA M3/6HCD FIN/
  DATA NER/6HERROR /
  DATA NERN/6HCRAP: /
  DATA ISTAR/3H***/
  DATA IRLK/'40/
  DATA IFN/3HEND/
  DATA NAM/6HCC    /
  MES(1)=NAM(1)
  MES(2)=NAM(2)
  CALL IO(MES)
  CALL CRACK(1,ISTAR,ISTAR)
  CALL IO1('0707)
  CALL IO1('0207)
  CALL IO1('0216)
  CALL IO('0203,300,LIS)
  CALL IO1('0210)
111  CALL IO('0701,27,MES)
      CALL IO1('0700)
      IF(IP(1).EQ.0)GOTO 111
      CALL IO1('0607)
      CALL IO('0602,27,MES)
      CALL IO1('0600)
      CALL IO1('0610)
      IF(IEN.EQ.MES(1))GO TO 1000
      CALL CRACK(60,MES;M)
      KO=0
      DO 20 J=1,300,3
      IF(MES(1).NE.LIS(J))GO TO 20
      KO=J
      M1=LIS(J+1)
      M2=LIS(J+2)
20   CONTINUE
      IF(KO)302,302,301
301  CONTINUE
      IF(M(3).EQ.ISTAR)GO TO 100
      IF(M(3).EQ.IBLK)GO TO 200

```

142

```

300  CONTINUE
    MES(1)=NFR(1)
    MES(2)=NFR(2)
303  CONTINUE
    CALL SQUASH(M,MES(3))
    CALL TO(MES)
    GO TO 111
302  MES(1)=NFRN(1)
    MES(2)=NFRN(2)
    GO TO 303
100  MES(1)=M1
    MES(2)=M2
    CALL SQUASH(M(2),MES(3))
    CALL SQ(MES)
    GO TO 111
200  MES(1)=M1
    MES(2)=M2
    I=2
    DO 201 J=3,10
    CALL MAKE(I,M(3),N,L)
    IF(L.NF.0)GO TO 300
201  MES(J)=N
    CALL SQ(MES)
    GO TO 111
1000 CALL IO1('0710)
    CALL TO(M3)
    RETURN
    END
    SUBROUTINE SQUASH(IS,ID)
    DIMENSION IS(24),ID(8)
    CALL PACK(24,IS,ID)
    RETURN
    END
    SUBROUTINE MAKE(I,M,N,L)
    DIMENSION M(24)
    DATA ICOM/'40/
    DATA MIN/'55/
    N=0
    L=0
    ISGN=1
1   IF(I.GT.60)GO TO 100
    IF(M(I).EQ.MIN)GO TO 900
    IF(M(I).EQ.ICOM)GO TO 100
    KK=M(I)-48
    IF(KK.LT.0.OR.KK.GT.9)L=1
    N=10*N+KK
    I=I+1
    GO TO 1
100  I=I+1
    N=N-ISGN
101  RETURN
900  I=I+1
    ISGN=-1
    IF(N.NF.0)L=1
    GO TO 1
    ENDS
$ASSIGN,6,17

```

143

\$CATALOG
TYPE=FG
NAME=CC
ASSIGN 2=NAMLIS,6=1,7=7
BEGIN
\$EDJ

144

```

$JOB NAMLIS
$ASSIGN 6 0
$REW,5
$INCLUDE IO
$INCLUDE FIO
$
  FORTRE-MENSION J(300),M(3)
    DATA IFN/3HEND/
    CALL OPN(7)
    L=0
1    CALL RDANW(7,3,M)
    IF(M(1).EQ.IEN)GOTO100
    L=L+1
    J(L)=M(1)
    L=L+1
    J(L)=M(2)
    L=L+1
    J(L)=M(3)
    GO TO 1
100  DO 200 LL=L,299
200  J(LL+1)=0
    CALL OPN(2)
    CALL REW(2)
    CALL WRANW(2,300,J)
    CALL CLOS(2)
    CALL EXIT
  ENDS

```

```

$ASSIGN 2 NAMLIS
$CATGO
BJ LOGGIN
EJ LOGOUT
AT ACOTAP
BS LAGSI2
CR CC
BB DLIM
BE DEFPNT
BF DFRAME
BK LDCNTR
BL DLANDM
DO DORBIT
BR DRATE
DS DSRATE
DT DEFTAP
DV* DVERS
EK ERADIR
ED EDAY
EL DELMRK
EX EXITWI
E1 ENH001
GL LNDMRK
IF LFRAME
IT LISTAP
KS SETKEY
LF FINFRM

```

145

LK LISDIR
LD LISLMK
LT FINDTP
MC USECUT
MF USEFIL
MG USEGRA
ML USELAP
MR USERAB
MS USESPA
NA NAVSYS
NC NAVCHG
NT XFORMS
PL SURPLT
QM MINBND
QP PEAKCD
QS SURFCO
RD ERTSLD
RF RESFRM
RK RESTDK
SF SETF
SK SAVEDK
SS STATUS
TB DEFMET
TD NRZDEC
TE FRMEAR
TS EARSAT
TT EARTAP
WC WCELL
WD FILDIM
WE FILELE
WI BAKGND
WL FILLIN
WP FILWGT
WT WHATAP
XF LDCNTV
ZA*ANNOT
ZD ANNDIS
ZE ANNINT
ZM*PLTMAP
ZZ LOGDUM
END
\$EDJ

146


```

$JOB LDCNTV
$REW,5
$ASSIGN,6,0
$INCLUDE SLEEP
$INCLUDE HIO
$INCLUDE IK
$INCLUDE LTV
$INCLUDE TVD
$INCLUDE WAIT
$INCLUDE AGE
$INCLUDE CSF
$INCLUDE LOOKUP
$INCLUDE STAGE
$INCLUDE TVSAT
$INCLUDE IO
$INCLUDE FIO
$FORTRAN
-----
      DIMENSION J(672),K(1120),L(672)
      DIMENSION MT(14),MF(14),MD(14)
      DIMENSION MESI(8);MESO(10),MESJ(10)
      DIMENSION MESE(24)
      DIMENSION ITAB(256)
      DIMENSION IGS(10)
      COMMON/SYSCOM/NDAT(100)
      DATA MFSI/6HLDLCNTV/
      DATA MFSO/6HDFRAME/
      DATA MFSJ/6HLFRAME/
      DATA MFSF/30HTV LOAD REQUEST REJECTED
      DATA IGS/0,1,2,4,8,16,32,63,0,0/
      CALL IO(MESI)
      DO 98 JA=1,256
98      ITAB(JA)=IK(JA/4)
C SOURCE AREA
      IA=MESI(1)
      CALL AGE(IA)
C FRAME
      IF=MESI(2)
      MESJ(3)=IF
      MESO(3)=IF
C REPEAT FACTOR
      NR=MESI(5)
C RELATIVE START
C COUNTING FROM 0
      IL=MESI(3)
      ITLS=1
      IF(IL.GE.0)GOTO 93
      ITLS=1-IL
      IL=0
93      IE=MESI(4)
      ITES=1
      IF(IE.GE.0)GOTO 94
      ITES=1-IF
      IE=0
94      ITLS=(ITLS-1)*NR+1
      ITES=(ITES-1)*NR+1
      IF(IA.IT.1)GOTO 100
      IF(IA.GT.8)GOTO 100

```

147

```

C GET AREA DIRFCTORY ENTRY
  CALL OPN(5)
  CALL RFW(5)
  CALL RFADW(5,112,J)
  IP=14*(IA-1)
  DO 1 JA=1,14
    IP=IP+1
1    MD(JA)=J(IP)
    CALL CLOS(5)
    IF(MD(1).LE.0)GOTO 100
C PERFORM TAPE LOOKUP
  CALL LOOK(2,MD(8),MF)
  IF(MT(1).EQ.0)GOTO 100
C LOOK UP FRAME ENTRY FOR PROJECT PROTECT KEY ONLY
  CALL LOOK(3,IF,MF)
  IF(MF(14).EQ.0)GOTO 2
  IF(MF(14).EQ.NDAT(1))GOTO 2
  IF(NDAT(95).LT.0)GOTO 2
  CALL SQ(MESO)
  GOTO 100
C TAPE
2  CONTINUE
  MESO(4)=MT(1)
C STARTING LINE OF TAPE
  ILF=MT(4)
C STARTING LINE OF AREA
  ILF=MD(3)-ILF+1
C OFFSET
  ILF=ILF+MD(5)*IL
  MESO(5)=ILF
C SAME WITH ELEMENTS
  IEF=MD(4)-MT(5)+1+MB(6)*IE
  MESO(6)=IEF
C NUMBER OF LINES
  MESO(9)=500/NR
C TV RELATIVE
  MESO(7)=ITLS
  MESO(8)=ITES
C DO ACTUAL LOAD
  MSEC=1000*(IA-1)+2*IL
C NSEC IS POINTER TO 5 LINE AREA ON DIGITAL DISK
  NSEC=MSEC
  CALL TVI(1)
  DO 96 JA=1,700
96  L(JA)=0
    JA=ITLS-1
    IF(ITLS.EQ.1)GOTO 97
    DO 95 KL=1,JA
95  CALL TVD(L)
C KL IS SCREEN LINE POINTER
97  KL=ITLS
    CALL OPN(4)
10  CALL SCRA(4,NSEC)
    CALL RFADW(4,1120,K)
    DO 11 JA=1,5
      JB=224*(JA-1)+1

```

148

```

      CALL CRACK(672,K(JB),J)
C JC IS SOURCE ELEMENT POINTER
      JC=IE
C KT IS SCREEN ELEMENT POINTER
      KT=ITES-1
14   DO 13 JD=1,NR
      KT=KT+1
      IF(KT.GT.700)GOTO 15
      JZ=J(JC)+1
13   L(KT)=ITAB(JZ)
      JC=JC+1
      IF(JC.GT.672)GOTO 15
      GOTO 14
15   DO 12 JB=1,NR
      CALL TVD(L)
      KL=KL+1
      IF(KL.EQ.500)GOTO 90
12   CONTINUE
11   CONTINUE
      NSEC=NSEC+10
      IF(NSEC.GT.MSEC+999)GOTO 90
      GOTO 10
90   DO 91 JA=4,10
91   MESJ(JA)=0
      CALL TVE
      CALL SQ(MESO)
      CALL SQ(MESJ)
      CALL EXIT
100  CALL TQ(MESE)
      CALL EXIT
      ENDS
$CATALOG
TYPE=FG
NAME=LDCNTV
ASSIGN 2=REELS,3=FRAMES,4=DK,5=DKDIR
BEGIN
$EOJ
$EOJ
/JFIN

NEXT JOB

```

149

PORN DATA 101
 DATA 0
 PIPK DATA '0202
 DATA 112
 PIBLOC ***
 POPK DATA '0201
 DATA 112
 DAC POB
 POB BLOK 112
 POBE DATA -1
 ENT DATA 90
 PCNT DATA 390

NET AVAILABLE TTY MESSAGES

*
 INTRTN TMA IDLE
 BOP INTEX
 TZM IDLE
 TME '200
 TZA
 TD4
 INTEX BUC 0,J

*
 WAITL CZM GOT
 BNZ ACTIV
 CZM IDLE
 BNN ACTIV

* 100 MILLISECOND WAIT

* (CALL TO INTERVAL TIMER MULTIPLEXOR)

TLO INTRTN
 TOI 100
 BLU '30
 TLO IDLE

* WAIT TILL ACTIVITY OR TIMEOUT RESETS IDLE FLAG
 BLU \$WAIT

* CALL TO SYSQ

ACTIV BLU '34
 TZM GOT

* INVOKE TTY IF CALLED FOR

TESTTY TMA IDLE
 KOB '212
 BNZ TWSC
 BLL \$CALLER
 DAC TTY
 DAC TTYP
 TZM IDLE
 TWSC CZM IDLE
 BOZ PINCK
 BON PINCK
 TMA IDLE
 TAM MES
 TLO CNTRL
 BLU '36

PINCK AUM GOT
 TLO PIB
 TFM IDLE
 BLU '27
 TKM PIBLOC
 TMA 0,K

150

```

      BOZ      PNOI
* DONT PUT RECORD IN QUE IF IT IS FULL
      TMA      PCNT
      SOA      1
      BON      PNOI
      TAM      PCNT
*
      TLO      PIP
      BLU      $I/O
      TLO      PIPK
      BLU      $I/O
      TNK      '0200
      BLU      $I/O
      TZM*     PIBLOC
      BLL      $BUMP2
      DAC      PIRN
      AUM      GOT
PNOI   CZM      POBE
      BOZ      INCK
      TMA      PIRN
      SMA      PORN
      BOZ      INCK
      TLO      POP
      BLU      $I/O
      TLO      POPK
      BLU      $I/O
      TNK      '0200
      BLU      $I/O
      BLL      $BUMP2
      DAC      PORN
      TZM      POBE
      AUM      GOT
INCK   CZM      IBF
      BOZ      NOI
* INPUT BUFFER IS FILL -- EMPTY IT
      TLO      IP
      BLU      $I/O
      TLO      IPK
      BLU      $I/O
      TNK      '0200
      BLU      $I/O
*      IRN=MOD(IRN+1,100)
      BLL      $BUMP
      DAC      IRN
      TZM      IBF
      AUM      GOT
* WAIT IF OUTPUT BUFFER IS FULL
NOI    CZM      OBE
      BOZ      CALTTO
* WAIT IF BUFFER IS EMPTY BUT NO RECORDS WAITING TO GO OUT
      TMA      IRN
      SMA      ORN
      BOZ      WAITL
* FILL OUTPUT BUFFER
      TLO      OP
      BLU      $I/O
      TLO      OPK
      BLU      $I/O

```

151

	TNK	'0200
	BLU	\$I/O
	BLL	\$BUMP
	DAC	ORN
	TZM	OBE
CAUTTO	TMA*	ASRFLG
	BON	WAITL
	TLO	TTYB
	BLU	\$I/O
	TFM	OBE
	AUM	CNT
	BUC	WAITL
* CODE LINKED BY BLU FROM PROGRAMS WHICH WILL FILL INPUT BUFFER		
PILLP	TJA	
	TKE	
	CZM	POBE
	BNN	**3
	TLO	POBE
	BLU	\$WAIT
	HXI	
	TAJ	
	TEK	
	TMA	'200
	UI1	
	RXI	
	AUM	PCNT
	TNI	112
PLDOP	TMA	POB+112,I
	TAM	0,K
	AOK	1
	BWI	PLOOP
	TZM	IDLE
	TFM	POBE
	BUC	RET
PILLI	TJA	
	TKE	
* WAIT IF TOO MANY RECORDS IN QUE		
	CZM	CNT
	BNN	**3
	TLO	CNT
	BLU	\$WAIT
* WAIT IF BUFFER ALREADY FULL		
	CZM	IBF
	BNN	**3
	TLO	IBF
	BLU	\$WAIT
* FORBID DISPATCHER		
	HXI	
	TAJ	
	TEK	
	TMA	'200
	UI1	
	RXI	
	TMA	CNT
	SOA	1
	AM	CN

152

```

      TFM      IBX
      TZM      IDLE
* COPY USERS OUTPUT TO IB
      TNI      LNG
ILOOP  TMA      0,K
      TAM      IB+LNG,I
      XOK      1
      BWI      ILOOP
      BUC      RET
* ENABLE DISINT AND GO BACK
RET     TMA      '200
      1
      XVE      '200
      TZA
      TD4
      BUC      0,J
* PLUG LINKAGES INT  BLU AREA AND DO FILE OPEN
START   TMA      INST1
      TAM*     L37
      TMA      INST3
      TAM*     L26
      TNK      '0107
      BLU      $I/O
      TNK      '0207
      BLU      $I/O
      HIT
      TOT      100
      RCT
      BLL      $CALLER
      DAC      PRINT
      DAC      PRINTP
MAIN    TMA      INST
      TAM*     INTRPT
      TMA      LVEL
      UA1
      UE1
      BUC      WAITL
LVEL    DATA   B17
INTRPT  DAC      '111
INST     BSL      $TVIR
INST1    BUL      FILLI
INST3    BUL      FILLP
L37      DAC      '37
L33      DAC      '33
L26      DAC      '26
      ENDS     START
$CATALOG
TYPE=RFG,PRIV
NAME=OPCOM,2
ASSIGN 1=1,2=OPMES
BE
GIN

```

153

```

$JOB H10
$ASSIGN,6,0
$ASSIGN,5,LR
$REW,5
$ASSEMB

      IDEN  H10
      XDEF  H10,H10
* CALLING SEQUENCE CU, CW, DW
IOCW  OCW   '00
IOBW  ODW   '00
H10   GAP   1
      TMA   0,I
      DMA   IOCW
      TAM   PLUG1
      TMA   0,I
      DMA   IODW
      TAM   PLUG2
      GAP   1
      HXI
      TMA   0,I
* NEGATIVE CW IMP DONT SEND
      BON   **3
PLUG1  ***
      RNZ   *-1
      GAP   1
      TMA   0,I
PLUG2  ***
      BNZ   *-1
      RXI
      BUC   0,J
      ENDS

$FILEMA
ESTAB,5,H10,0,1,0,4
EXIT
$EOJ

```

154


```
JOB AGE
ASSIGN 5 LR
REW,5
ASSIGN,6,0
FORTRAN
      SUBROUTINE AGE(N)
      COMMON/SYSCOM/NDAT(100)
      DO 1 J=1,8
      K=J+82
      NDAT(K)=NDAT(K)+1
      K=82+N
      NDAT(K)=0
      RETURN
      ENDS
FILEMA
TAB,5,AGE,0,1,0,4
IT
OJ
```

155

```

$JOB GETGAM
$ASSIGN,6,0
$ASSIGN 5 LR
$REW,5
$FORTRA
    SUBROUTINE GETGAM(ID,IT,BETA,BDOT)
    DIMENSION J(560)
    DIMENSION MER(24)
    COMMON/SYSCOM/NDAT(100)
    DATA MER/30HNO GAMMA SHIFTS AVAILABLE /
    DATA NSECT/-1/
    NSEC=5*MOD(ID,100)
    IF(NSEC.EQ.NSECT)GOTO 90
    IF(NDAT(92).LT.0)CALL WAIT(NDAT(92))
    NDAT(92)=-1
    CALL OPN(9)
    CALL SCRA(9,NSEC)
    CALL READW(9,560,J)
    CALL CLOS(9)
    NDAT(92)=0
90  NSECT=NSFC
    BETA=0.
    BDOT=0.
    IHIT=0
    DO 1 JA=1,560,7
    IF(J(JA).NE.ID)GOTO 1
    KIND=J(JA+6)/4096
    IF(KIND.NE.14)GOTO 1
    IHIT=IHIT+1
    IF(IHIT.GT.1)GO TO 80
    BETA=J(JA+2)
    BDDT=J(JA+3)
80  IF(J(JA+1).GT.IT)GO TO 1
    BETA=J(JA+2)
    BDDT=J(JA+3)
1   CONTINUE
    IF(IHIT.EQ.0)CALL TQ(MER)
    BETA=BETA/100.
    BDDT=BDOT/100.
    RETURN
    ENDS
$FILEMA
ESTAB,5,GETGAM,0,1,0,4
EXIT
$EOJ

```

156

```

$JOB TVSAT
$ASSIGN,6,0
$ASSIGN,5,LR
$REW,5
$OPTION 9
$FORTRAN

```

```

SUBROUTINE TVSAT(IF,ILT,IET,IL,IE,IT,ID)

```

```

C INPUTS

```

```

C IF -- FRAME NUMBER
C ILT -- TV CURSOR LINE
C IET -- TV CURSOR ELEMENT

```

```

C OUTPUTS

```

```

C IL -- SSCC LINE
C IE -- SSCC ELEMENT
C IT -- FRAME START TIME

```

```

C M1 IS REEL DEF

```

```

C REEL
C YYDDD
C HHMMSS

```

```

C LS
C ES

```

```

C M2 IS FRAME DEF

```

```

C FRAME
C REEL
C REC ST
C ELE ST
C TVL
C TVE
C MAG

```

```

DIMENSION M1(14),M2(14)
DIMENSION NER1(24),NER2(24)
COMMON/SYSCOM/NDAT(100)
DATA NER1/30HFRAME NON-EXIST
DATA NER2/30HTAPE NON-EXIST
CALL LOOK(3,IF,M2)
IF(M2(1).EQ.0.OR.M2(2).EQ.0)GO TO 800
CALL LOOK(2,M2(2);M1)
IF(M1(1).EQ.0.OR.M1(2).EQ.0)GO TO 900

```

```

C YYDDD

```

```

ID=M1(2)

```

```

C HHMMSS

```

```

IT=M1(3)

```

```

C LINE

```

```

IL=ILT-M2(5)
IL=2*IL
ISIZ=M2(7)
LINST=M1(4)
LINST=LINST+M2(3)-1
MAG=1
IF(ISIZ.LT.116)MAG=3
IF(ISIZ.LT.56)MAG=6
IF(ISIZ.GT.115)MAG=1
IL=IL/MAG+LINST

```

157

```

C ELEM(VTT=M1(5)-1+M2(4)
      IE=IET-M2(6)
      IE=6*IF
      IE=IE/MAG+IEST
      RETURN
800  CALL TO(NER1)
      GOTO 1000
900  CALL TO(NER2)
1000 IL=0
      IE=0
      IT=0
      ID=0
      RETURN
      END
      SUBROUTINE LOOK(LUN,KEY,MES)
C LOOKUP AND SAVE
C   LUN=2 == REELS
C   LUN=3 == FRAMES
      DIMENSION MES(14),M(300),ILOC(3)
      COMMON/SYSCOM/NDAT(100)
      DATA ILOC/0,94,93/
      DATA N/0/
      IF(NDAT(54).NE.0)N=0
      IF(N.EQ.0)GO TO 100
      DO 1 J=1,N,15
      IF(M(J+1).NE.KEY)GO TO 1
      IF(M(J).NE.LUN)GO TO 1
      GO TO 200
1     CONTINUE
100  IF(N.EQ.300)N=0
      M(N+1)=LUN
      IF(LUN.NE.2.AND.LUN.NE.3)STOP TVSAT
      I=ILOC(LUN)
      IF(NDAT(I).LT.0)CALL WAIT(NDAT(I))
      NDAT(I)=-1
      CALL LOOKUP(LUN,KEY,M(N+2))
      NDAT(I)=0
      J=N+1
      N=N+15
200  DO 201 I=1,14
      J=J+1
201  MES(I)=M(J)
      RETURN
      ENDS
$FILEMA
ESTAB,5,TVSAT,0,1,0,4
EXIT
$EOJ

```

158

```

$JOB LOOKUP
$ASSIGN 6 0
$OPTIONS 8 23
$ASSIGN 5 LR
$REW,5
$FORTRAN

```

```

C SUBROUTINE LOOKUP(LUN,KEY,MUNG)
  FOR LOOKING UP AN ENTRY IN FRAMES OR REELS FILE
  DIMENSION MUNG(14)
  COMMON/LOKP/NSEC,MUN,JSEC(112)
  CALL OPN(LUN)
  CALL REW(LUN)
  NSEC=-1
  MUN=LUN

```

```

C NUMBER OF ENTRIES IN FILE

```

```

N=JARY(1)-1
C DISTANCE FROM START OF FILE TO FIRST DATA ENTRY
IOFF=JARY(2)
DO 1 J=1,14
1  MUNG(J)=0
DO 2 J=10,N
2  IF(JARY(J).EQ.KEY)GO TO 3
CONTINUE
GO TO 100
3  ISEC=IOFF+J/8
  IREL=1+14*MOD(J,8)
  CALL SCRA(LUN,ISEC)
  CALL READW(LUN,112,JSEC)
  LST=IREL+13
  JA=0
  DO 4 J=IREL,LST
4  JA=JA+1
  MUNG(JA)=JSEC(J)
100 CALL CLOS(LUN)
RETURN
END

```

```

FUNCTION JARY(J)
C FUNCTION JARY (J) IS THE JTH ENTRY IN THE FILE INDEX
C PERFORMS IO IF NECESSARY

```

```

COMMON/LOKP/NSEC,MUN,JSEC(112)
MSEC=(J-1)/112
MWOR=MOD(J-1,112)+1
IF(NSEC.FQ.MSEC)GO TO 1
NSEC=MSEC
CALL SCRA(MUN,MSEC)
CALL READW(MUN,112,JSEC)
1 JARY=JSEC(MWOR)
RETURN
END$

```

```

$FILEMA
ESTAR,5,LOOKUP,0,1,0,4
EXIT
$EOJ

```

159

```
$JOB VERS
$REW,5
$INCLUDE IO
$INCLUDE FIO
$ASSIGN 6 0
$FORTRAN
    DIMENSION J(112)
    DATA J/112*0/
1   FORMAT(10A3)
    READ(7,1)(J(L),L=1,10)
    CALL OPN(2)
    CALL SCRA(2,0)
    CALL WRITW(2,112,0)
    CALL CLOS(2)
    CALL EXIT
    ENDS
$FILEMA
EXIT
$ASSIGN 2=VERS
$CATGO
MCYDAS VERS S2.10H1.00
$EOJ
```

160

\$JOB TTY
\$ASSIGN 6,0
\$REW,5
\$INCLUDE,CSF
\$INCLUDE,STAGE
\$INCLUDE,IO
\$FORTRAN

```
      SUBROUTINE QUIT5
      COMMON/SYSCOM/NDAT(100)
      CALL EXIT
      RETURN
      END
      SUBROUTINE SQUASH(IS,ID)
      DIMENSION IS(24),ID(8)
      CALL PACK(24,IS,ID)
      RETURN
      END
      SUBROUTINE MAKE(I,M,N,L)
      DIMENSION M(24)
      DATA MIN/'55/
      DATA ICOM/'40/
      N=0
      L=0
      ISGN=1
1      IF(I.GT.60)GO TO 100
      IF(M(I).FQ.MIN)GO TO 900
      IF(M(I).FQ.ICOM)GO TO 100
      KK=M(I)-48
      IF(KK.LT.0.OR.KK.GT.9)L=1
      N=10*N+KK
      I=I+1
      GO TO 1
100    I=I+1
      N=N+ISGN
101    RETURN
900    I=I+1
      IF(N.NE.0)L=1
      ISGN=-1
      GO TO 1
      END
      DIMENSION NE5(24)
      DIMENSION LIS(300)
      DIMENSION MES(27);M(60)
      DIMENSION NERN(2)
      DIMENSION NER(2)
      COMMON/SYSCOM/NCOM(100)
      DATA NE5/30HMUST LOGGIN FIRST
      DATA NER/6HERROR /
      DATA NERN/6HCRAP: /
      DATA ISTAR/3H***/
      DATA ICOM/3H,,,/
      DATA IRLK/'40/
      CALL CRACK(1,ISTAR,ISTAR)
      CALL CRACK(1,ICOM,ICOM)
111    CALL IO(64*1+'7)
      CALL IO(64*1+1,27;MES)
```

161

```

CALL IO1(64*1)
CALL IO1(64*1+'10)
CALL CRACK(60,MES,M)
CALL IO1(64*2+'7)
CALL IO(64*2+'17,0,0)
CALL IO(64*2+1,300,LIS )
CALL IO1(64*2+'10)
KO=0
DO 20 J=1,300,3
  IF(MES(1).NE.LIS(J))GO TO 20
  KO=J
  M1=LIS(J+1)
  M2=LIS(J+2)
20  CONTINUE
  IF(KO)302,302,301
301  CONTINUE
C  FORCES LOGGIN PROCEDURE
  IF(NCOM(1).EQ.0.AND.KO.NE.1)CALL TO(NE5)
  IF(NCOM(1).EQ.0.AND.KO.NE.1)CALL QUITs
  IF(M(3).EQ.1STAR)GO TO 100
  IF(M(3).EQ.1BLK) GO TO 200
300  CONTINUE
  MES(1)=NFR(1)
  MES(2)=NFR(2)
303  CONTINUE
  CALL SQUASH(M,MES(3))
  CALL TO(MES)
  CALL QUITs
302  CALL OC(MES)
  CALL QUITs
100  MES(1)=M1
  MES(2)=M2
  CALL SQUASH(M(4),MES(3))
  CALL SQ(MES)
  CALL QUITs
200  MES(1)=M1
  MES(2)=M2
  I=2
  DO 201 J=3,10
  CALL MAKE(I,M(3),N,L)
  IF(L.NE.0)GO TO 300
201  MES(J)=N
  CALL SQ(MES)
  CALL QUITs
ENDS

```

\$ASSEMB

```

XDEF  OC,OC
OC    GAP  1
      TJM  RET
      TIK
      BLU  '25
      BUC* RET
RET   ***
      ENDS

```

```

$CATALOG
TYPE=FG
NAME=TTY
$ASSIGN 1=1,2=NAMLIS
NOMAP
BEGIN
$EOJ

```

162


```

$JOB NRZL
$ASSIGN,5,LR
$ASSIGN,6,0
$REW,5
$FORTRAN

```

```

SUBROUTINE NRZDEC(LINE,L,IFERROR)
DIMENSION LINE(2);IIB(96),I(28)
DATA KS,LEVELB,LEVELT,ILOOK/-10,75,100,85/
IB(N)=IIR(N)
LASBIT=0
IERROR=0

```

```

C KLUDGE TO TRY TO GET MORE DECODES
E CANT USE MORE THAN LINE AND TIME THEN

```

```

ILOOK=72
DO 2 I=1,ILOOK
I2=2*I
I1=I2-1
N=LINE(I1)
M=LINE(I2)
NN=1
MM=1
IF(N.LT.LEVELT)NN=0
IF(M.LT.LEVELT)MM=0
IF(N.GT.LEVELB.AND.N.LT.LEVELT)NN=MM
IF(M.GT.LEVELB.AND.M.LT.LEVELT)MM=NN
IF(MM.EQ.NN)GO TO 1
IERROR=1
RETURN

```

```

1 J=1
IF(LASBIT.EQ.MM)J=0

```

```

IIB(I)=J
LASBIT=MM

```

```

2 CONTINUE

```

```

L(1)=2*IB(35+KS)+IB(36+KS)
L(2)=8*IB(37+KS)+4*IB(38+KS)+2*IB(39+KS)+IB(40+KS)
L(3)=8*IB(41+KS)+4*IB(42+KS)+2*IB(43+KS)+IB(44+KS)
L(4)=8*IB(45+KS)+4*IB(46+KS)+2*IB(47+KS)+IB(48+KS)
L(5)=2*IB(51+KS)+IB(52+KS)
L(6)=8*IB(53+KS)+4*IB(54+KS)+2*IB(55+KS)+IB(56+KS)
L(7)=4*IB(58+KS)+2*IB(59+KS)+IB(60+KS)
L(8)=8*IB(61+KS)+4*IB(62+KS)+2*IB(63+KS)+IB(64+KS)
L(9)=4*IB(66+KS)+2*IB(67+KS)+IB(68+KS)
L(10)=8*IB(69+KS)+4*IB(70+KS)+2*IB(71+KS)+IB(72+KS)
L(11)=4*IB(75+KS)+2*IB(76+KS)+IB(77+KS)
L(12)=2*IB(81+KS)+IB(82+KS)
L(13)=16*IB(81+KS)+8*IB(82+KS)+4*IB(83+KS)+2*IB(84+KS)+IB(85+KS)
L(14)=4*IB(88+KS)+2*IB(89+KS)+IB(90+KS)
L(15)=2*IB(91+KS)+IB(92+KS)
L(16)=16*IB(86+KS)+8*IB(87+KS)+4*IB(88+KS)+2*IB(89+KS)+IB(90+KS)
L(17)=4*IB(91+KS)+2*IB(92+KS)+IB(93+KS)
L(18)=2*IB(94+KS)+IB(95+KS)
L(19)=16*IB(91+KS)+8*IB(92+KS)+4*IB(93+KS)+2*IB(94+KS)+IB(95+KS)
L(20)=IB(97+KS)
L(21)=IB(98+KS)
L(22)=IB(99+KS)
L(23)=IB(100+KS)

```

163

```

L(24)=4*IB(18+KS)+2*IB(19+KS)+IB(20+KS)
L(25)=8*IB(21+KS)+4*IB(22+KS)+2*IB(23+KS)+IB(24+KS)
L(26)=IB(73+KS)
L(27)=IB(74+KS)
L(28)=IB(65+KS)
RETURN
5ND
SUBROUTINE NRZL(LINE,ITIME)
DIMENSION J(100),K(300),L(28)
CALL -FADW(2,100,J)
CALL CRACK(300,J,K)
CALL NRZDEC(K,L,IE)

```

8

```

LINE=1000*L(1)+100*L(2)+10*L(3)+L(4)

```

8

```

90  ITIME=10*ITIME+L(JJ)
    RETURN
1   LINE=0
    ITIME=0
    RETURN
ENDS

```

```

$FILEMA
ESTAB,5,NRZL,0,1,0,4
EXIT
$EOJ

```

164

```
$JOB FRAMES
$OPTIONS 0 8 23
$REW,5
$ASSIGN 6,0
$INCLUDE IO
$INCLUDE FIO
$INCLUDE CSF
$FORTRAN
    DIMENSION J(1120)
    DO 10 K=1,1120
10   J(K)=0
        1 =10
    T-L III OPN(2)
        2)
    CALL III-IIIW(2,1120,J)
    CALL CLOS(2)
    CALL EXIT
    ENDS
$ASSIGN 2 FRAMES
$CATGO
$EOJ
```

165

```
$JOB ZLANDM
$REW,5
$ASSIGN,6,0
$ASSIGN 2=LANDMA
$INCLUDE FIO
$INCLUDE IO
$FORTRA
    DIMENSION JDIR(560)
    DATA JDIR/560*0/
    CALL OPN(2)
    N=0
    DO 1 J=1,100
    CALL SCRA(2,N)
    CALL WRITW(2,560,JDIR)
1   N=N+5
    CALL EXIT
    ENDS
$CATGO
$EOJ
```

166

```
$JOB REELS
$OPTIONS 0 8 23
$REW,5
$ASSIGN 6,0
$INCLUDE IO
$INCLUDE FIO
$INCLUDE CSF
$FORTRAN
    DIMENSION J(1120)
    DO 10 K=1,1120
10  J(K)=0
    J(1)=10
    CALL OPN(2)
    CALL RFW(2)
    CALL WRITW(2,1120;J)
    CALL CLOS(2)
    CALL EXIT
    ENDS
$ASSIGN 2 REFLS
$CATGO
$EOJ
```

167

```

$JOB KLUDGE
$ASSIGN 6 0
$REW,5
$FORTRAN
    SUBROUTINE DOIT(LUN)
    DIMENSION J(112)
    CALL OPN(LUN)
    CALL RFW(LUN)
    CALL READW(LUN,112,0)
    J(2)=10
    CALL SCRA(LUN,0)
    CALL WRITW(LUN,112,0)
    CALL CLOS(LUN)
    RETURN
    END
    DO 10 K=2,3
10   CALL DOIT(K)
    CALL EXIT
    ENDS
$INCLUDE IO
$INCLUDE FIO
ASSIGN 2=REELS,3=FRAMES
$CATGO
$ASSIGN 6,1
$EOJ

```

168

```
$JOB SETKEY
$ASSIGN,6,0
$REW,5
$INCLUDE CSF
$FORTRAN
  DIMENSION MES(10)
  COMMON/SYSCOM/NDAT(100)
  DATA MES/6HSETKEY/
  CALL IO(MES)
  KEY=MES(1)
  KSTAT=MES(2)
  IF(KSTAT.NE.1.AND.KSTAT.NE.0)CALL ABORT
  IF(KEY.LT.0.OR.KEY.GT.22)CALL ABORT
  N=1
  IF(KEY.GT.0)N=2**KEY
  M=1777777777.XOR.N
  NDAT(95)=NDAT(95).AND.M
  N=N*KSTAT
  NDAT(95)=NDAT(95).OR.N
  CALL EXIT
END$
$CATALOG
TYPE=FG
NAME=SETKEY
BEGIN
$EOJ
```

169

```

$JOB DPROJ
$ASSIGN 6 0
$INCLUDE FIO
$INCLUDE IO
$FORTRAN
    DIMENSION N(112)
    DATA N/112*0/
    CALL OPN(2)
    CALL SCRA(2,0)
    I=0
1   FORMAT(15,1X,08)
10  READ(7,1)I1,I2
    IF(I1.FQ.9999)GO TO 100
    I=I+1
    J=2*(I-1)      +1
    N(J)=I1
    N(J+1)=I2
    GO TO 10
100 CALL WRITW(2,112,N)
    CALL EXIT
    ENDS

```

\$ASSIGN 2 PROJS

\$CATGO

```

1000 00000002
1210 00000002
1220 00000002
1230 00000002
1250 00000002
1500 00000002
1510 00000000
1520 00000002
1600 00000002
1610 00000002
1620 00000002
1630 00000002
1640 00000002
1800 00000002
1900 00000002
3910 00000002
4439 00000002
5600 00000002
    00 00000002
6700 00000002
6900 00000002
6999 40000000
9999 00000000

```

\$EDJ

170


```

$JOB DEFTAP
$REW,5
$INCLUDE CSF
$INCLUDE FIO
$INCLUDE IO
$ASSIGN 6 0
$INCLUDE WAIT
$FORTRAN
    DIMENSION NAM(2)
    DIMENSION JDIR(1120)
    DIMENSION MES(8),JSEC(112)
    DIMENSION MER(24),NER(24)
    COMMON/SYSCOM/IP(100)
    DATA MER/30HNOT AUTHORIZED TO DT
    DATA NER/30HTAPE REEL FILE FULL
    DATA NAM/6HDEFTAP/
    CALL WAIT(IP(94))
    IP(94)=-1
    CALL OPN(2)
199  CALL RFW(2)
    CALL READW(2,1120;JDIR)
1  MES(1)=NAM(1)
    MES(2)=NAM(2)
    CALL IO(MES)
    IKEY=IP(95).AND.'40000000
    IF(IKEY.NE.0)GO TO 6999
    CALL CLOS(2)
    IP(94)=0
    CALL TO(MER)
    CALL EXIT
6999 IF(NAM(1).EQ.MES(1))GO TO 299
    IF(JDIR(1).EQ.1000)GO TO 999
    KEY=KFIND(JDIR,MES(1))
        2 10+
    CALL ,GZE\(. ,11K#ZT(=)
    IST=1+14*MOD(KEY,8)
    LST=IST+7
    DO 99 JAY=IST,LST
        INDX=1+JAY-IST
99  JSEC(JAY)=MES(INDX)
    CALL SCRA(2,10+KEY/8)
    CALL WRITW(2,112,JSEC)
100 CALL SCRA(2,0)
    CALL WRITW(2,1120;JDIR)
    GO TO 199
299 CALL CLOS(2)
    IP(54)=1
    IP(94)=0
    CALL EXIT
999 CALL TO(NER)
    GO TO 299
END
FUNCTION KFIND(J,K)
    DIMENSION J(1120)

```

171

```
$JOB DFRAME
$ASSIGN 6 0
$REW,5
$INCLUDE CSF
$INCLUDE FIO
$INCLUDE IO
$INCLUDE WAIT
$FORTRAN
```

```
    DIMENSION NAM(2)
    DIMENSION JDIR(1120)
    DIMENSION MES(8),JSEC(112)
    DIMENSION NER(24),NERR(24)
    COMMON/SYSCOM/NDAT(100)
    DATA NAM/6HDFRAME/
    DATA NERR/30HERROR -- FRAME LOCK
    DATA NFR/30HFRAME FILE FULL
    CALL WAIT(NDAT(93))
    NDAT(93)=-1
    CALL OPN(2)
199  CALL REW(2)
    CALL READW(2,1120;JDIR)
1    MES(1)=NAM(1)
    MES(2)=NAM(2)
    CALL IO(MES)
    IF(NAM(1).EQ.MES(1))GO TO 299
    IF(JDIR(1).EQ.1000)GO TO 999
    KEY=KFIND(JDIR,MES(1))
    CALL SCRA(2,10+KEY/8)
    CALL READW(2,112,JSEC)
    IST=1+14*MOD(KEY,8)
    LST=IST+7
    NP=JSEC(IST+13)
C PRIVILEGED PROJECT
    IKEY=NDAT(95).AND.'40000000
    IF(IKEY.NE.0)GO TO 98
C UNRESTRICTED FRAME
    IF(NP.EQ.0)GO TO 98
C PROJECT MATCH
    IF(NP.EQ.IPROJ)GO TO 98
    CALL TO(NERR)
    GO TO 199
98  DO 99 JAY=IST,LST
    INDX=1+JAY-IST
99  JSEC(JAY)=MES(INDX)
    CALL SCRA(2,10+KEY/8)
    CALL WRITW(2,112,JSEC)
100 CALL SCRA(2,0)
    CALL WRITW(2,1120;JDIR)
    GO TO 199
299 CALL CLOS(2)
C SIGNAL SYSTEM CHANGE WORD
    NDAT(54)=1
C RELEASE FRAME LOCK
    NDAT(93)=0
```

172

```

CALL EXIT
999 CALL TO(NER)
GO TO 299
END
FUNCTION KFIND(J,K)
DIMENSION J(1120)
IF(J(1).EQ.10)GO TO 100
N=J(1)-1
DO 2 L=10,N
IF(J(L).EQ.K)GO TO 200
2 CONTINUE
100 KFIND=J(1)
J(KFIND)=K
J(1)=J(1)+1
RETURN
200 KFIND=I
RETURN
END$

```

```

$CATALOG
TYPE=FG
NAME=DFRAME
ASSIGN 2=FRAMES
BEGIN
$EOJ

```

```

$JOB WHATAP
$REW,5
$INCLUDE CSF
$INCLUDE IO
$INCLUDE FIO
$INCLUDE WAIT
$ASSIGN,6,0
$FORTRAN
    DIMENSION JDIR(1120),MES(8),MOUT(24)
    COMMON/TLIS/NLIS,NDAY(1000)
    COMMON/SYSCOM/NDAT(100)
    DATA MFS/6HWHATAP/,
    NLIS=0
    CALL IO(MES)
    CALL WAIT(NDAT(94))
    NDAT(94)=-1
    CALL OPN(2)
    CALL REW(2)
    CALL READW(2,1120;JDIR)
    N=JDIR(1)-1
    NSEC=JDIR(2)
    DO 100 J=10,N,80
    CALL SCRA(2,NSEC+J/8)
    CALL READW(2,1120;JDIR)
    DO 100 K=1,80
    L=2+(K-1)*14
100  CALL Q(JDIR(L))
    CALL CLOS(2)
    NDAT(94)=0
    DO 200 J=1,NLIS
    ENCODE(72,222,MOUT)NDAY(J)
200  CALL TO(MOUT)
    CALL EXIT
222  FORMAT(I10)
    END
    SUBROUTINE Q(N)
    COMMON/TLIS/NLIS,NDAY(1000)
    IF(N.EQ.0)RETURN
    IF(NLIS.NE.0)GO TO 1
    NLIS=1
    NDAY(1)=N
1    DO 2 J=1,NLIS
    IF(NDAY(J).EQ.N)RETURN
    IF(NDAY(J).GT.N)GO TO 3
2    CONTINUE
    NLIS=NLIS+1
    NDAY(NLIS)=N
    RETURN
3    DO 4 K=J,NLIS
    L=NLIS+J-K
4    NDAY(L+1)=NDAY(L)
    NLIS=NLIS+1
    NDAY(J)=N
    RETURN
    ENDS

```

174

```

$CATALOG
TYPE=FG,PRIV
NAME=WHATAP
ASSIGN 2=REELS
BEGIN
$EOJ
$REW,5
$INCLUDE FIO
$INCLUDE IO
$INCLUDE CSF
$ASSIGN 6 0
$FORTRAN
    DIMENSION J(112),MES(10)
    COMMON/SYSCOM/NDAT(100)
    DATA MFS/6HDVERS /
    CALL OPN(2)
    CALL SCRA(2,0)
    CALL READW(2,112,0)
    CALL SCRA(2,0)
    CALL IO(MES)
    DO 1 K=1,8
1      J(K)=MFS(K)
    IF(NDAT(1).NE.1520)CALL ABORT
    CALL WRITW(2,112,0)
    CALL CLOS(2)
    CALL EXIT
    ENDS
$CATALOG
TYPE=FG
NAME=DVERS
ASSIGN 2=VERS
BEGIN
$EOJ

```

175

```

$JOB RESFRM
$ASSIGN 6 0
$REW,5
$INCLUDE CSF
$INCLUDE FIO
$INCLUDE IO
$INCLUDE WAIT
$FORTRAN
    DIMENSION NAM(2)
    DIMENSION JDIR(1120)
    DIMENSION MES(8),JSEC(112)
    DIMENSION NERR(24)
    COMMON/SYSCOM/NDAT(100)
    DATA NAM/6HRESFRM/
    DATA NERR/30HERROR -- FRAME LOCK
    CALL WAIT(NDAT(93))
    NDAT(93)=-1
    CALL OPN(2)
199  CALL REW(2)
    CALL READW(2,1120;JDIR)
1  MES(1)=NAM(1)
    MES(2)=NAM(2)
    CALL IQ(MES)
    IF(NAM(1).EQ.MES(1))GO TO 299
    IKEY=NDAT(95).AND.'40000000
    IF(IKEY.NE.0)GO TO 98
    CALL TO(NERR)
    GO TO 199
98  CONTINUE
    I1=MES(1)
    I2=MES(2)
        DO 777 JA=I1,I2
    KEY=KFIND(JDIR,JA)
    CALL SCRA(2,10+KEY/8)
    CALL READW(2,112,JSEC)
    CALL SCRA(2,10+KEY/8)
    IST=1+14*MOD(KEY,8)
    JSEC(IST)=MES(3)
777 CALL WRITW(2,112,JSEC)
100 CALL SCRA(2,0)
    CALL WRITW(2,1120;JDIR)
    GO TO 199
299 CALL CLOS(2)
    NDAT(93)=0
    CALL EXIT
    END
    FUNCTION KFIND(J,K)
    DIMENSION J(1120)
    IF(J(1).EQ.10)GO TO 100
    N=J(1)-1
    DO 2 L=10,N
    IF(J(L).EQ.K)GO TO 200
2  CONTINUE
100 KFIND=J(1)
    J(KFIND)=K
    J(1)=J(1)+1

```

+13

176

200 RETURN
KFIND=L
RETURN
END\$

\$CATALOG
TYPE=FG
NAME=RESFRM
ASSIGN 2=FRAMES
BEGIN
\$EDJ

177

```

$JOB LOGDUM
$REW,5
$INCLUDE IO
$ASSIGN 6 0
$INCLUDE FIO
$INCLUDE CSF
$FORTRAN
    DIMENSION MES(112),MOUT(24)
    DATA MFS/6HLOGDUM/
    CALL IQ(MES)
    IF(MES(1).NE.987898)CALL EXIT
    CALL OPN(2)
    CALL RFW(2)
    CALL READW(2,112,MES)
    DO 10 J=1,112;2
    IF(MES(J).EQ.0)GO TO 10
    F=MES(J+1)
    F=F/60.
    ENCODE(72,222,MOUT)MES(J),F
222  FORMAT('PROJECT',15,' ',F8.2,' MINUTES')
    CALL TO(MOUT)
10   CONTINUE
    DO 11 J=1,112
11   MES(J)=0
    CALL SCRA(2,0)
    CALL WRITW(2,112,MES)
    CALL CIOS(2)
    CALL EXIT
    ENDS
$CATALOG
TYPE=FG
NAME=LOGDUM
ASSIGN 2=LOG
BEGIN
$EOJ

```



```
$JOB PRNOUT
$REW,5
$ASSIGN 6,0
$INCLUDE CSF
$FORTRAN
    DIMENSION MES(10);NAM(2)
    DIMENSION MOUT(24)
    DATA NAM/6HPRNOUT/
1    MES(1)=NAM(1)
    MES(2)=NAM(2)
    CALL IQ(MES)
    IF(MES(1).EQ.NAM(1))CALL EXIT
    J=MES(1)
    J=J+1
    ENCODE(72,222,MOUT)(MES(K),K=2,J)
    CALL TQ(MOUT)
    GO TO 1
222  FORMAT(7I8)
    ENDS
$CATALOG
NAME=PRNOUT
TYPE=FG
BEGIN
$EOJ
```

```
$JOB ERADIR
$REW,5
$INCLUDE CSF
$INCL DE IO
$INCLUDE FIO
$ASSIGN 6,0
$FORTRA
    DIMENSION MES(10);J(112)
    DATA MFS/6HERADIRZ
    DATA J/112*0/
    CALL IQ(MES)
    CALL OPN(2)
    CALL RFW(2)
    CALL WRITW(2,112,0)
    CALL CLOS(2)
    CALL EXIT
END$
$CATALOG
TYPE=FG
NAME=ERADIR
ASSIGN 2=DKDIR
BEGIN
$EDJ
```

180

```

$JOB NRZDEC
$ASSIGN,6,0
$REW,5
$INCLUDE IO
$INCLUDE FIO
$INCLUDE STAGE
$INCLUDE CSF
$INCLUDE NRZI
$INCLUDE FTIME
$INCLUDE ITIME
$FORTRA
    FUNCTION IDEC(N)
    DATA I7/'60/
    NN=N
    N1=MOD(N,10)
    NN=NN/10
    N2=MOD(NN,10)
    NN=NN/10
    N3=MOD(NN,10)
    N1=N1+1Z
    N2=N2+1Z
    N3=N3+1Z
    N2=N2.SHIFT.8
    N3=N3.SHIFT.16
    IDEC=N1.OR.N2.OR.N3
    RETURN
    END
    DIMENSION M1(24),M2(24)
    DIMENSION MES(8)
    COMMON/SYSCOM/ICOM(100)
    DATA M1/30HTAPE START      TIME
    DATA M2/30H      PERCENT SUCCESSFUL DECODES
    DATA MES/6HNRZDEC/
    CALL IQ(MES)
    CALL OPN(2)
    CALL REW(2)
    IQ=0
    ID=-1
    LIMIT=MES(1)
    IF(LIMIT.GT.2000)LIMIT=2000
    DO 1 M=1,LIMIT
    IF(ICOM(1).EQ.0)GOTO 222
    CALL NRZI(LINE,ITM)
    IF(LINE.EQ.0)GOTO 1
    IG=IG+1
    NID=LINE+1-M
    IF(ID.EQ.NID)GOTO 1
    ID=NID
    T=FTIME(ITM)
    X=.6*FLOAT(LINE)/3600.
    T=T-X
    ITI=ITIME(T)
    M1(5)=IDEC(NID)
    M1(10)=IDEC(ITI)
    M1(9)=IDEC(ITI/1000)
    CALL TO(M1)

```

181

```

1  CONTINUE
   Y=LIMIT
   X=IG
   X=X/Y
   IX=100*X
   M2(1)=IDFC(IX)
   CALL TO(M2)
222 CALL RFW(2)
   CALL CLOS(2)
   CALL EXIT
200 FORMAT('FRACTION OF SUCCESSFUL DECODES = ',E5,3)
ENDS

```

```

$CATALOG
TYPE=FG
NAME=NRZDEC
ASSIGN 2=11
BEGIN
$EOJ

```

182

```

$JOB ACQTAP
$ASSIGN,6,0
$REW,5
$INCLUDE FIO
$INCLUDE IO
$INCLUDE CSF
$INCLUDE NRZI
$INCLUDE STAGE
$INCLUDE FTIME
$INCLUDE ITIME
$FORTRA

```

```

    DIMENSION MES(10);MOUT(10),MER(24)
    COMMON/SYSCOM/ICOM(100)
    DATA MFS/6HACQTAP/
    DATA MOUT/6HDEFTAP/
    DATA MER/30HNO SUCCESSFUL DECODES
    CALL IQ(MES)
    CALL OPN(2)
    CALL REW(2)
    DO 1 J=1,1000
    IF(ICOM(1).EQ.0)GOTO 222
    CALL NRZI(L,IT)
    IF(L.EQ.0)GOTO 1
    T=FTIME(IT)
    X=.6*FLOAT(L)/3600.
    T=T-X
    MOUT(3)=MES(1)
    MOUT(4)=MES(2)
    MOUT(5)=ITIME(T)
    MOUT(6)=L-J+1
    MOUT(7)=1
    MOUT(8)=0
    MOUT(9)=0
    MOUT(10)=0
    CALL SQ(MOUT)
    ENCODE(72,100,MER)MOUT
    GOTO 2
1    CONTINUE
2    CALL TQ(MER)
222  CALL REW(2)
    CALL CLOS(2)
    CALL EXIT
100  FORMAT(2A3,8I7)
    ENDS

```

```

$CATALO
NAME=ACQTAP
TYPE=FG
ASSIGN*2=11
BEGIN
$EOJ

```

183

\$JOB STATUS
\$ASSIGN 6,0
\$REW,5
\$INCLUDE CSF
\$FORTRA

```
FUNCTION IOCT(N)
  DATA M1,M2,M3/'7,'170,'700/
  DATA I7/3H000/
  N1=M1.AND.N
  N2=M2.AND.N
  N3=M3.AND.N
  N2=N2.SHIFT.5
  N3=N3.SHIFT.10
  IOCT=I7.OR.N1.OR.N2.OR.N3
  RETURN
END
```

```
SUBROUTINE MAIN(N)
  DIMENSION N(2)
  DIMENSION MES(10)
  DATA MES/6HSTATUS/
  CALL IO(MES)
  CALL PRINT(N(1))
```

```
  M=1
1  MM=II(N(17),M)
  IF(MM.EQ.0)RETURN
  MM=MM+16
  CALL PRINT(N(MM))
  M=MM
  GO TO 1
END
```

```
FUNCTION II(N,M)
  STARTS AT LOC N LOOKING FOR WORD M
  DIMENSION N(2)
```

```
  I=1
10  IF(N(I).EQ.M+128)GOTO 100
  IF(I.GT.500)GOTO 101
  I=I+8
  GO TO 10
100 II=1
  RETURN
101 II=0
  RETURN
END
```

```
SUBROUTINE PRINT(N)
  DIMENSION N(8)
  DIMENSION MOUT(24)
  DATA I1/3H /
  DATA N2/3H 00/
  MOUT(1)=N(2)
  MOUT(2)=N(3)
  DO 1 J=3,24
1  MOUT(J)=IB
  J=N(4).SHIFT.-18
  J1=J.AND.7
  J2=J.AND.'70
  J2=J2.SHIFT.5
  J=N2.OR.J1.OR.J2
```

184

```
MOUT(4)=J
MOUT(5)=IOCT(N(4)/512)
MOUT(6)=IOCT(N(4))
MOUT(8)=IOCT(N(6)/512)
MOUT(9)=IOCT(N(6))
NNN=N(7)
MOUT(11)=IOCT(NNN/512)
MOUT(12)=IOCT(NNN)
MOUT(14)=IOCT(N(8))
CALL TO(MOUT)
RETURN
ENDS
```

```
$ASSEMB
START  BLL  $MAIN
        DAC  '201
        BLU  $EXI
        ENDS START
```

```
$CATALOG
TYPE=FG,PRIV
NAME=STATUS
BEGIN
$EOJ
```

```
$JOB SETF
$ASSIGN,6,0
$REW,5
$INCLUDE CSF
$FORTRAN
    DIMENSION MES(8)
    COMMON/SYSCOM/NDAT(100)
    DATA MES/6HSETF /
    CALL IQ(MES)
    J=NDAT(56).AND.'3Z7Z0000
    J=J+MES(1)
    NDAT(56)=J
    CALL EXIT
    ENDS
$CATALOG
TYPE=FG
NAME=SETF
BEGIN
$EOJ
```

186


```
$JOB DLIM
$ASSIGN,6,0
$REW,5
$INCLUDE CSF
$FORTRAN
    DIMENSION MES(8)
    COMMON/SYSCOM/NDAT(100)
    DATA MES/6HDLIM /
    CALL IO(MES)
C LOWER LIMIT+4096*UPPER LIMIT
  NDAT(57)=MES(1)+4096*MES(2)
  CALL EXIT
END$
$CATALOG
TYPE=FG
NAME=DLIM
BEGIN
$EOJ
```

187

```

$JOB DRATE
$ASSIGN,6,0
$REW,5
$INCLUDE CSF
$FORTRA
    DIMENSION MES(8)
    COMMON/SYSCOM/NDAT(100)
    DATA MES/6HDRATE /
    CALL IQ(MES)
C ARGUMENT IS NUMBER OF FRMS PER SECOND TO LOG
    IF(MES(1).GT.30)CALL EXIT
    IF(MES(1).LE.0)CALL EXIT
    X=MES(1)
    X=1./X
    X=X*1000.
    NDAT(58)=X
    CALL EXIT
END$
$CATALOG
TYPE=FG
NAME=DRATE
BEGIN
$EOJ

```

188

```

$JOB LOOP
$ASSIGN,6,0
$REW,5
$INCLUDE CSF
$FORTRA
    DIMENSION MES(8),NAM(2)
    COMMON/SYSCOM/NDAT(100)
    DATA NAM/6HLOOP /
    DATA MFS/6HLOOP /
    CALL IQ(MES)
    IF(NDAT(1).EQ.0)CALL EXIT
1   MES(1)=NAM(1)
    MES(2)=NAM(2)
    CALL IQ(MES)
    IF(MES(1).NE.NAM(1))CALL EXIT
    IF(NDAT(1).EQ.0)CALL EXIT
    CALL SLEEP(NDAT(58))
    IT=NDAT(57)/4096
    IB=NDAT(57)-4096*IT
    ID=MOD(NDAT(56),4096)
    L=IT-IB
    IF(ID.EQ.IT)GOTO 2
    NDAT(56)=NDAT(56)+1
    GOTO 1
2   NDAT(56)=NDAT(56)*L
    GOTO 1
END$

```

\$ASSEMBR

```

    IDEN  SLEE
    XDEF  SLEEP,SLEEP
SLEEP   GAP  1
        TJM  RET
        TMI  0,I
        TLO  RTN
        BLU  '30
        TFM  K
        TLO  K
        BLU  $WAI
        BUC* RET
RET     ***
RTN     TZM  K
        TME  '200
        TZA
        TD4
        BUC  0,J
K       ***
        END$

```

```

$CATALOG
TYPE=FG
NAME=LOOP
BEGIN
$EOJ
$JOB NAVFRM
$ASSIGN,6,0
$REW,5
$INCLUDE CSF
$INCLUDE FIO
$INCLUDE IO

```

189

\$INCLUDE LOOKUP
\$INCLUDE TVSAT
\$INCLUDE WAIT
\$FORTRA

C CALLS NAVIGATION SYSTEM FOR DAY OF CURRENT FRAME

DIMENSION MIN(8),MOUT(10)
COMMON/SYSCOM/NDAT(100)
DATA MIN/6HNAVFRM/
DATA MOUT/6HNAVSY/
CALL IQ(MIN)
IF=NDAT(56).AND.'Z777'
CALL TVSAT(IF,100,100,1,J,IT,ID)
MOUT(3)=ID
CALL SQ(MOUT)
CALL EXIT
ENDS

\$CATALOG
TYPE=FG
NAME=NAVFRM
%ASSIGN 2=REELS,3=FRAMES
BEGIN
\$EOJ

```
$JOB DEFPNT
$ASSIGN,6,0
$REW,5
$INCLUDE CSF
$FORTRA
    DIMENSION MES(8)
    COMMON/SYSCOM/NDAT(100)
    DATA MES/6HDEFPNT/
    CALL IQ(MES)
    NDAT(59)=MES(1)
    NDAT(60)=MES(2)
    CALL EXIT
    ENDS
$CATALOG
TYPE=FG
NAME=DEFPNT
BEGIN
$EOJ
```

```
$JOB DORBIT
$REW,5
$ASSIGN,6,0
$INCLUDE CSF
$FORTRA
  DIMENSION MIN(8),MOUT(10)
  DATA MIN/6HDORBIT/
  DATA MOUT/6HDLANDM/
C ORBITS ARE ENCODED AS 'LANDMARKS' OF TYPE 12 AND 13
  CALL IQ(MIN)
  MOUT(3)=MIN(1)
  MOUT(4)=0
  MOUT(5)=0
  MOUT(8)=MIN(2)
  MOUT(9)=MIN(3)
  MOUT(10)=12
  MOUT(6)=MIN(4)
  MOUT(7)=MIN(5)
  CALL SQ(MOUT)
  MOUT(10)=13
  MOUT(8)=MIN(6)
  MOUT(9)=MIN(7)
  MOUT(6)=MIN(8)
  MOUT(7)=0
  CALL SQ(MOUT)
  CALL EXIT
END$
$CATALOG
TYPE=FG
NAME=DORBIT
BEGIN
$EOJ
```

192

```
JOB DSRATE
ASSIGN,6,0
REW,5
INCLUDE CSF
FORTRA
    DIMENSION MIN(8),MOUT(10)
    DATA MIN/6HDSRATE/
    DATA MOUT/6HDLANDM/
$PIN RATE IS ENCODED AS 'LANDMARK' OF KIND 11
    CALL IO(MIN)
    MOUT(3)=MIN(1)
    MOUT(4)=0
    MOUT(5)=0
    MOUT(6)=0
    MOUT(7)=0
    MOUT(8)=MIN(2)
    MOUT(9)=0
    MOUT(10)=11
    CALL SQ(MOUT)
    CALL EXIT
END$
$CATALOG
TYPE=FG
NAME=DSRATE
BEGIN
$EOJ
```

193

```

$JOB DLANDM
$ASSIGN 6 0
$REW,5
$INCLUDE CSF
$INCLUDE IO
$INCLUDE FIO
$INCLUDE WAIT
$FORTRA

```

```

SUBROUTINE SRT(J)
  DIMENSION J(560)
  DO 1 K=1,560,7
  DO 1 L=K,560,7
  JL=J(L+1)
  JK=J(K+1)
  IF(JL.EQ.0)JL=240100
  IF(JK.EQ.0)JK=240100
  IF(JK.LE.JL)GOTO 1
  DO 2 M=1,7
  MK=M-1+K
  ML=M-1+L
  LTEM=J(MK)
  J(MK)=J(ML)
  J(ML)=LTEM
2  CONTINUE
1  CONTINUE
RETURN
END

```

```

  DIMENSION NAM(2)
  DIMENSION MES(8)
  DIMENSION JDIR(560)
  DIMENSION NER3(24),NER9(24)
  COMMON/SYSCOM/NDAT(100)

```

```

C DEFINES LANDMARKS AND OTHER PICTURE RELATED DATA
C INPUT IS SYDDDD HHMMSS KEY P1,P2,P3,P4, KIND
C ONLY ONE ENTRY OF UNIQUE SYDDDD HHMMSS KIND
C FOR EXAMPLE, FOR LANDMARKS ...
C INPUT IS SYDDDD, HHMMSS, KEY, SATLIN, SATELE, LAT, LON

```

```

DATA NER9/40HPICTURE INFO FILE UPDATE REJECTED

```

```

DATA NAM/6HDLANDM/

```

```

DATA NER3/40HNO ROOM IN PICTURE DATA FILE

```

```

199 MES(1)=NAM(1)
  MES(2)=NAM(2)
  CALL IO(MES)
  IF(MES(1).EQ.NAM(1))GO TO 299
  IF(MES(1).EQ.0)CALL ABORT
  IKEY=NDAT(95).AND.'40000000
  IF(IKEY.NE.0)GO TO 191
  CALL TO(NER9)
  CALL EXIT
191 CONTINUE
  ID=MES(1)
  IT=MES(2)
  IK=MES(3)
  IL=MES(4)
  IE=MES(5)
  ILAT=MES(6)

```

194


```

KIND=MFS(8)
CALL WAIT(NDAT(92))
NDAT(92)=-1
CALL OPN(4)
MSEC=5*MOD(ID,100)
CALL SCRA(4,MSEC)
CALL READW(4,560,JDIR)
DO 11 K=1,560,7
  IF(JDIR(K).NE.ID)GOTO 11
  IF(JDIR(K+1).NE.IT)GOTO 11
  KAY=JDIR(K+6)/4096
  IF(KAY.NE.KIND)GOTO 11
  GOTO 50
11 CONTINUE
DO 10 K=1,560,7
  IF(JDIR(K).EQ.0)GOTO 50
10 CONTINUE
CALL TO(NER3)
CALL CLOS(4)
NDAT(92)=0
CALL EXIT
C SYDDDD
50 JDIR(K)=ID
  K=K+1
C HHMMSS
60 JDIR(K)=IT
  K=K+1
C LAT
  IF(ILAT.NE.0)JDIR(K)=ILAT
  IF(KIND.GT.9)JDIR(K)=ILAT
  K=K+1
C LONG
  IF(ILON.NE.0)JDIR(K)=ILON
  IF(KIND.GT.9)JDIR(K)=ILON
  K=K+1
C LINE
  IF(IL.NE.0)JDIR(K)=IL
  IF(KIND.GT.9)JDIR(K)=IL
  K=K+1
  IF(IE.NE.0)JDIR(K)=IE
  IF(KIND.GT.9)JDIR(K)=IE
  K=K+1
  JDIR(K)=IK+4096*KIND
C
CALL SRT(JDIR)
CALL SCRA(4,MSEC)
CALL WRITW(4,560,JDIR)
CALL CLOS(4)
NDAT(92)=0
GO TO 199
299 CALL EXIT
END$

```

```

$CATALOG
TYPE=FG
NAME=DLANDM
$ASSIGN 4=LANDMA
BEGIN
$EOJ

```

195

```

$JOB FINDTP
$ASSIGN 6,0
$INCLUDE IO
$INCLUDE FIO
$INCLUDE CSF
$INCLUDE WAIT
$FORTRAN
    DIMENSION JDIR(1120),MFS(8),MOUT(24)
    DIMENSION NAM(2)
    DIMENSION MTY(24)
    COMMON/SYSCOM/NDAT(100)
    DATA NAM/6HFINDTP/
    DATA MOUT/6HLISTAR/
    DATA MTY/30HNO TARE REELS
    CALL WAIT(NDAT(94))
    NDAT(94)=-1
    CALL OPN(2)
199  MES(1)=NAM(1)
    MES(2)=NAM(2)
    CALL IQ(MES)
    IF(MES(1).EQ.NAM(1))GO TO 299
    CALL RFW(2)
    KN=0
    CALL READW(2,1120;JDIR)
    N=JDIR(1)-1
    IF(MES(1).EQ.0)GO TO 299
    DO 100 J=10,N,80
    CALL SCRA(2,10+J/8)
    CALL READW(2,1120;JDIR)
    DO 100 K=1,80
    L=1+(K-1)*14
    IF(JDIR(L+1).NE.MES(1))GO TO 100
    MOUT(3)=JDIR(L)
    DO 90 L=4,10
90  MOUT(L)=0
    KN=KN+1
    CALL SQ(MOUT)
100  CONTINUE
    IF(KN.EQ.0)CALL TQ(MTY)
    GO TO 199
299  CALL CLOS(2)
    NDAT(94)=0
    CALL EXIT
    ENDS

```

```

$CATALOG
TYPE=FG
NAME=FINDTP
ASSIGN 2=REELS
BEGIN
$EOJ

```

196

```

$JOB LISTAP
$REW,5
$ASSIGN 6,0
$INCLUDE LOOKUP
$INCLUDE,FIO
$INCLUDE IO
$INCLUDE CSF
$INCLUDE WAIT
$FORTRAN
    DIMENSION MES(8),MOUT(24),NAM(2),JSEC(14)
    COMMON/SYSCOM/NDAT(100)
    DATA NAM/6HLISTAP/
1    MES(1)=NAM(1)
    MES(2)=NAM(2)
    CALL IO(MES)
    IF(MES(1).EQ.NAM(1))CALL EXIT
    DO 10 J=1,8
    IF(MES(J).EQ.0)GO TO 10
    IF(NDAT(94).LT.0)CALL WAIT(NDAT(94))
    NDAT(94)=-1
    CALL LOOKUP(2,MES(J),JSEC)
    NDAT(94)=0
    IF(JSEC(1).EQ.0)GO TO 9
    ENCODE(72,222,MOUT)(JSEC(K),K=1,5)
222  FORMAT('T',2I7,18;2I6)
    CALL TO(MOUT)
    GO TO 10
9    ENCODE(72,333,MOUT)MES(J)
333  FORMAT('NO TAPE',I7)
    CALL TO(MOUT)
10   CONTINUE
    GO TO 1
    ENDS
$CATALOG
TYPE=FG
NAME=LISTAP
ASSIGN 2=REELS
BEGIN
$EOJ

```

```
$JOB FINFRM
$ASSIGN 6 0
$REW,5
$INCLUDE CSF
$INCLUDE IO
$INCLUDE FIO
$INCLUDE WAIT
$FORTRAN
    SUBROUTINE LT(N)
        DIMENSION MES(10)
        DATA MFS/6HLISTAP/
        MES(3)=N
        DO 1 J=4,10
1      MES(J)=0
        CALL SQ(MES)
        RETURN
    END
    SUBROUTINE LF(N)
        DIMENSION MES(10)
        DATA MFS/6HLFRAME/
        MES(3)=N
        DO 1 J=4,10
1      MES(J)=0
        CALL SQ(MES)
        RETURN
    END
    DIMENSION IFRAME(208)
    DIMENSION JDIR(1120),ITAPE(100),IOUT(100)
    DIMENSION MER1(24)
    DIMENSION MES(10)
    COMMON/SYSCOM/NDAT(100)
    DATA NT/0/
    DATA MER1/30HNO FRAMES
    DATA NF/0/
    DATA MFS/6HFINFRM/
    CALL WAIT(NDAT(94))
    NDAT(94)=-1
    CALL OPN(2)
    CALL RFW(2)
    CALL READW(2,1120;JDIR)
    NTAPE=JDIR(1)-1
    CALL IQ(MES)
    DO 100 J=10,NTAPE;80
    CALL SCRA(2,10+J/8)
    CALL READW(2,1120;JDIR)
    DO 100 K=1,80
    L=1+(K-1)*14
    IF(JDIR(L+1).NE.MES(1))GO TO 100
    NT=NT+1
    ITAPE(NT)=JDIR(L)
    IOUT(NT)=0
100  CONTINUE
    CALL CLOS(2)
    NDAT(94)=0
    CALL WAIT(NDAT(93))
    NDAT(93)=-1
```

198

```

CALL OPN(3)
CALL RFW(3)
CALL READW(3,1120;JDIR)
NFRAME=JDIR(1)-1
DO 200 J=10,NFRAME,80
CALL SCRA(3,10+J/8)
CALL READW(3,1120;JDIR)
DO 200 K=1,80
L=1+(K-1)*14
DO 201 I=1,NT
IF(ITAPE(I).EQ.JDIR(L+1))GO TO 202
201 CONTINUE
GO TO 200
202 NF=NF+1
IFRAME(NF)=JDIR(L)
IF(IOUT(I).EQ.0)CALL LT(ITAPE(I))
IOUT(I)=1
200 CONTINUE
CALL CLOS(3)
NDAT(93)=0
IF(NF.EQ.0)GO TO 400
DO 300 J=1,NF
300 CALL LF(IFRAME(J))
CALL EXIT
400 CALL TO(MER1)
CALL EXIT
END$

$CATALOG
TYPE=FG
NAME=FINFRM
ASSIGN 2=REELS,3=FRAMES
BEGIN
$EOJ

```

```

$JOB SAVEDK
$ASSIGN,6,0
$REW,5
$INCLUDE CSF
$INCLUDE FIO
$INCLUDE IO
$FORTRAN
    DIMENSION J(2240)
    DIMENSION MES(8)
    DATA MFS/6HSAVEDK%
    CALL IO(MES)
    CALL OPN(2)
    CALL OPN(3)
    CALL OPN(4)
    CALL RFW(2)
    CALL RFW(3)
    CALL RFW(4)
    CALL READW(3,112,J)
    CALL CLOS(3)
    CALL WRITW(2,112,J)
    DO 1 N=1,400
    CALL READW(4,2240,J)
1   CALL WRITW(2,2240,J)
    CALL RFW(2)
    CALL RFW(4)
    CALL CLOS(2)
    CALL CLOS(4)
    CALL EXIT
    ENDS
$CATALOG
TYPE=FG
NAME=SAVEDK
ASSIGN 2=11,3=DKDIR,4=DK
BEGIN
$EOJ

```

200

```

$JOB RESTDK
$ASSIGN,6,0
$REW,5
$INCLUDE FIO
$INCLUDE CSF
$INCLUDE IO
$FORTRA
    DIMENSION J(2240)
    DIMENSION MES(8),LK(10)
    DATA LK/6HLISDIR/
    DATA MES/6HRESTDK/
    CALL OPN(2)
    CALL RFW(2)
    CALL OPN(3)
    CALL RFW(3)
    CALL OPN(4)
    CALL RFW(4)
    CALL IO(MES)
    CALL READW(2,112,J)
    CALL WRITW(3,112,J)
    CALL RFW(3)
    CALL CLOS(3)
    CALL SQ(LK)
    DO 1 N=1,400
    CALL READW(2,2240;J)
1   CALL WRITW(4,2240;J)
    CALL RFW(2)
    CALL RFW(4)
    CALL CLOS(2)
    CALL CLOS(4)
    CALL EXIT
    ENDS
$CATALOG
TYPE=FG
NAME=RESTDK
ASSIGN 2=11,3=DKDIR,4=DK
BEGIN
$EOJ

```

201

\$JOB DELMRK
\$ASSIGN 6 0
\$REW,5
\$INCLUDE WAIT
\$FORTRAN

DIMENSION MES(8),JDIR(560)
DIMENSION MER(24)
COMMON/SYSCOM/NDAT(100)
DATA MFR/30HUNAUTHORIZED EI ***
DATA MES/6HDELMRK/
CALL IO(MES)
IF(MES(1).EQ.0)CALL ABORT
CALL WAIT(NDAT(92))
NDAT(92)=-1
CALL OPN(2)
IKEY=NDAT(95).AND.'40000000'
IF(IKEY.NE.0)GO TO 6999
CALL TO(MER)
CALL CLOS(2)
NDAT(92)=0
CALL EXIT

6999 MSEC=5*MOD(MES(1),100)
CALL SCRA(2,MSEC)
CALL READW(2,560,JDIR)
DO 1 J=1,560,7
IF(JDIR(J).NE.MES(1))GOTO 1
IF(JDIR(J+1).NE.MES(2))GOTO 1
KIND=JDIR(J+6)/4096
IF(MES(3).NE.KIND)GOTO 1
JDIR(J)=0
1 CONTINUE
CALL SCRA(2,MSEC)
CALL WRITW(2,560,JDIR)
CALL CLOS(2)
NDAT(92)=0
CALL EXIT
END\$

\$INCLUDE CSF
\$INCLUDE IO
\$INCLUDE FIO
\$CATALOG
TYPE=FG
NAME=DELMRK
ASSIGN 2=LANDMA
BEGIN
\$EQJ

202


```

$JOB WCELL
$ASSIGN,6,0
$REW,5
$INCLUDE CSF
$INCLUDE FIO
$INCLUDE IO
$FORTRAN
    DIMENSION J(560),MES(8),MOUT(24)
    COMMON/SYSCOM/NDAT(100)
    DATA MFS/6HWCELL /
    CALL IO(MES)
    IF(NDAT(92).LT.0)CALL WAIT(NDAT(92))
    NDAT(92)=-1
    CALL OPN(2)
    CALL REW(2)
    MB=0
    MN=0
    DO 1 M=1,100
    CALL READW(2,560,0)
    MM=0
    DO 2 LI=1,560,7
    IF(J(LI).EQ.0)GOTO 2
    MM=MM+1
2    CONTINUE
    IF(MM.LT.MB)GOTO 1
    MB=MM
    MN=M-1
1    CONTINUE
    CALL CLOS(2)
    NDAT(92)=0
    ENCODE(72,333,MOUT)MN,MB
333  FORMAT('CELL',I3,' HAS',I4,' ENTRIES')
    CALL TO(MOUT)
    CALL EXIT
    ENDS
$INCLUDE WAIT
$CATALOG
TYPE=FG
NAME=WCELL
ASSIGN 2=LANDMA
BEGIN
$EOJ

```

```

$JOB ABSLD
$ASSIGN,6,0
$REW,5
$INCLUDE CSF
$INCLUDE EDGFCO
$INCLUDE FIO
$INCLUDE FTIME
$INCLUDE GETGAM
$INCLUDE IO
$INCLUDE LOOKUP
$INCLUDE SATFAR
$INCLUDE STAGE
$INCLUDE WAIT
$FORTRAN
    DIMENSION IS(8193)
    DIMENSION IT(2731)
    DIMENSION ID(1120)
    DIMENSION MES(8)
    DIMENSION MUNG(14)
    DIMENSION MOUT(24)
    COMMON/SYSCOM/ICOM(100)
    EQUIVALENCE(IT(1),IS(5462))
    DATA MFS/6HABSLD /
    DATA MOUT/30HTAPE LOAD COMPLETED
    CALL OPN(2)
    CALL OPN(3)
    CALL IO(MES)
C MES(1)= DISK AREA 1 - 8
    NSEC=MFS(1)
C MES(2)=STARTING ELEMENT TAPE RELATIVE
    IES=MES(2)
C MES(3) IS LINE INCREMENT
    ILD=MES(3)
    IF(ILD.EQ.0)ILD=1
C MES(4) IS ELEMENT INCREMENT
    IED=MES(4)
    IF(IED.EQ.0)IED=3
C MES(5) IS SATELLITE LINE TAPE IS NOW POSITIONED AT
    IPL=MES(5)
C MES(6) IS FIRST SAT ELEMENT ON TAPE
    IOF=MES(6)
C MES(7) IS FEC KEY
    KEE=MES(7)
C MES(8) IS REFL NUMBER
    IF(ICOM(94).LT.0)CALL WAIT(ICOM(94))
    ICOM(94)=-1
    CALL LOOKUP(4,MES(8),MUNG)
    ICOM(94)=0
    IDAY=MUNG(2)
    PST=FTIME(MUNG(3))
    CALL GETGAM(IDAY,MUNG(3),BFTA,BDOT
    NSEC=(NSEC-1)*1000
    NR=500*ILD
    NE=672*IFD
    IEL=IES+NE-1

```

204

```

      M=M+1
      DO 99 J=2,6
        ISEC(M)=MES(J)
99    M=M+1
E TAPENO
      ISEC(M)=M1(1)
      CALL RFW(3)
      CALL WRITW(3,112,ISEC)
      CALL CLOS(3)
      CALL SQ(MOUT)
      CALL EXIT
900  CALL TO(MER)
      CALL EXIT
      ENDS
$CATALOG
TYPE=FG,PRIV
NAME=LDCNTR
ASSIGN 2=11,3=DKDIR,4=REELS
BEGIN
$EOJ

```

205

```

      I=J
      IO=NDAT(J+82)
1    CONTINUE
      CALL AGE(I)
      NMES(2)=NMES(2)+I
      CALL TO(NMES)
      N=i
      RETURN
      END
      DIMENSION MES(8),M1(14),MOUT(10),MER(24),ISEC(112)
      COMMON/SYSCOM/NDAT(100)
      DATA MFS/6HLDCNTR/
      DATA MFR/30HLDCNTR PARAMETER ERROR
      DATA MOUT/6HABSLD /
      CALL IQ(MES)
      CALL WAIT(NDAT(94))
      NDAT(94)=-1
      CALL LOOKUP(4,MES(1),M1)
      NDAT(94)=0
      IF(M1(1).EQ.0)GO TO 900
      LS=MES(2)-M1(4)+1
      IF(LS.LT.1)GO TO 900
      IES=MES(3)-M1(5)+1
      IF(IES.LF.0)GO TO 900
C FIRST TAPE RELATIVE ELEMENT TO LOAD
      MOUT(4)=IES
C LINE INCREMENT
      IF(MES(4).EQ.0)MES(4)=1
      MOUT(5)=MES(4)
C ELEMENT INCREMENT
      IF(MES(5).EQ.0)MES(5)=3
      MOUT(6)=MES(5)
C SATELLITE LINE
      MOUT(7)=MES(2)
C FIRST SATELLITE LMENT ON TAPE
      MOUT(8)=M1(5)
C EEC KEY
      MOUT(9)=MES(6)
C TAPE NUMBER
      MOUT(10)=M1(1)
      CALL PICK(NSEC)
C AREA TO PUT DATA
      MOUT(3)=NSEC
      CALL POS(LS,M1(4),M1(3))
      CALL OPN(3)
      CALL RFW(3)
      CALL READW(3,112,ISEC)
      M=14*(NSEC-1)+1
C SYYDDD
      ISEC(M)=M1(2)
      M=M+1
C HHMMSS
      ISEC(M)=M1(3)

```

206

```

$JOB LDCNTR
$ASSIGN 6,0
$REW,5
$INCLUDE CSF
$INCLUDE IO
$INCLUDE FIO
$INCLUDE LOOKUP
$INCLUDE WAIT
$INCLUDE AGE
$INCLUDE STAGE
$INCLUDE NRZL
$INCLUDE FTIME
$FORTRA

```

```

      SUBROUTINE POS(M,LINE,ITIME )
      DIMENSION DUMMY(2)
      COMMON/SYSCOM/ICOM(100)
      DIMENSION MOUT(24)
      DATA MOUT/40HTAPE IDENTITY VERIFIED BY LDCNTR
      DATA KEY/0/
      CALL OPN(2)
      CALL RFW(2)
      N=M-1
      IF(N.LT.1)RETURN
      T=FTIME(ITIME)
      DO 1 J=1,N
      IF(ICOM(1).EQ.0)GOTO 222
      IF(KEY.EQ.0)GOTO 100
      CALL READW(2,2,DUMMY)
      GOTO 1
100  CALL NRZL(L,IT)
      IF(L.EQ.0)GOTO 1
      I=L-J
      IF(IABS(I-LINE).GT.5)GOTO 1
      X=FTIME(IT)
      O=.6*FLOAT(L)/3600.
      X=X-O
      Y=ABS(X-T)*3600.
      IF(Y.GT.100.)GOTO 1
      KEY=1
      CALL TO(MOUT)
1    CONTINUE
      CALL CLOS(2)
      RETURN
222  CALL RFW(2)
      CALL CLOS(2)
      CALL EXIT
      END
      SUBROUTINE PICK(N)
      DIMENSION NMES(24)
      COMMON/SYSCOM/NDAT(100)
      DATA NMES/30HAREA 0 USED BY LDCNTR
      I=1
      IO=0
      DO 1 J=1,8
      IF(IO.GT.NDAT(J+82))GO TO 1

```

```
333 ENCODE(36,333,MMOUT,M4,M3,M2,M1
    FORMAT('SYSTEM IDLE AT ',4I1)
    CALL TO(MOUT)
    CALL EXIT
900 DO 901 J=1,112
901 JLOG(J)=0
    GO TO 3
ENDS
```

```
$CATALOG
TYPE=FG
NAME=LOGOUT
ASSIGN 2=LOG,3=SYSCOM
BEGIN
$EOJ
```

```

$JOB LOGOUT
$ASSIGN 6 0
$REW,5
$INCLUDE CSF
$INCLUDE FIO
$INCLUDE IO
$FORTRAN
COMMON/SYSCOM/NCOM(100)
DIMENSION MCOM(100)
DIMENSION MES(8),MOUT(24),JLOG(112)
DIMENSION MMOUT(12)
DIMENSION MESX9(10)
EQUIVALENCE(MOUT(12),MMOUT(1))
DATA MFS/6HLOGOUT/
DATA MFSX9/6HDOI0XX/
  CALL FROGS(7,MES)
  CALL IO(MES)
  IF(NCOM(1).EQ.0)CALL EXIT
  KPROJ=NCOM(1)
  NCOM(1)=0
  DO 983 JA=1,100
983  MCOM(JA)=NCOM(JA)
  CALL OPN(3)
  CALL RFW(3)
  CALL WRITW(3,100,MCOM)
  CALL CLOS(3)
  NCOM(76)=1
  CALL SQ(MESX9)
  NCLOCK=NCOM(97)
  CALL OPN(2)
  CALL RFW(2)
  IF(MES(1).EQ.987898)GO TO 900
  CALL READW(2,112,JLOG)
3  CONTINUE
  CALL SCRA(2,0)
  DO 2 J=1,112,2
  IF(JLOG(J).EQ.0.OR.JLOG(J).EQ.KPROJ)GO TO 100
2  CONTINUE
  J=111
100 JLOG(J)=KPROJ
  JLOG(J+1)=JLOG(J+1)+NCLOCK
  CALL WRITW(2,112,JLOG)
  CALL CLOS(2)
  F60=NCLOCK
  F60=F60/60.
  ENCODE(72,222,MOUT)KPROJ,F60
222 FORMAT('PROJECT',I5,F8.2,' MINUTES')
  NCOM(96)=NCOM(96)+NCOM(97)
  IF(NCOM(96).GT.86400)NCOM(96)=NCOM(96)-86400
  NCOM(97)=0
  MM=NCOM(96)/60
  MM=MOD(MM,60)
  M1=MOD(MM,10)
  M2=MOD(MM/10,10)
  MM=NCOM(96)/3600
  M3=MOD(MM,10)
  M4=MM/10

```

```

$JOB ENH001
$ASSIGN,6,0
$REW,5
$INCLUDE CSF
$INCLUDE HIO
$INCLUDE IK
$INCLUDE IO
$INCLUDE FIO
$FORTRA
    COMMON/SYSCOM/NDAT(100)
    DIMENSION MES(8),M(64)
    DATA MFS/6HENH001%
    CALL IO(MES)
    CALL OPN(2)
    CALL RFW(2)
    CALL READW(2,64,M)
    CALL RFW(2)
    IF(MES(1).NE.0)GOTO 100
    DO 1 J=1,64
1      M(J)=0
100    D=MES(2)-MES(1)
        IF(D.LT.1.)D=1.
        S1=MES(4)-MES(3)
        S2=MES(6)-MES(5)
        S3=MES(8)-MES(7)
        S1=S1/D
        S2=S2/D
        S3=S3/D
        X1=MES(3)
        X2=MES(5)
        X3=MES(7)
        I=MES(1)
        J=MES(2)
        DO 200 L=1,J
            M1=X1
            M2=X2
            M3=X3
            X1=X1+S1
            X2=X2+S2
            X3=X3+S3
200    M(L)=4096*M1+64*M2+M3
        CALL WRITW(2,64,M)
        CALL CLOS(2)
        IF(J.NE.63)CALL EXIT
        CALL HIO('101','2140','1000000)
        CALL HIO('101','2000','23000000+IK(M(1)))
        DO 300 L=2,64
300    CALL HIO('101','2000','21000000+IK(M(L)))
        CALL HIO('101','2140',NDAT(75))
        CALL EXIT
    ENDS
$CATALOG
TYPE=FG
NAME=ENH001
ASSIGN 2=ENH1
BEGIN
$FILEMA
CREATE,ENH1,0,1,3,1
EXIT
$EOJ

```

210


```
1    CONTINUE
C IM SHOULD COUNT FROM 1
  IF (IS.NE.-1) IM=IM+1
  RETURN
END$
$FILEMA
ESTAB,5,ADDR,0,1,0,4
EXIT
$EOJ
```

\$JOB ADDR
\$ASSIGN 6 0

\$Z\$W,GN 5 LR
\$FORTRA

```
      SUBROUTINE ADDR(ID,IT,IL,IE,IDL,IDE,IS,IM)
C      ***** INPUTS *****
C ID IS DAY NUMBER
C IL IS SATELLITE START LINE
C IE IS SATELLITE START ELEMENT
C      ***** OUTPUTS *****
C IL IS CHANGED TO LIE ON A LINE IN QUE
C IE IS CHANGED TO LIE ON WORD BOUNDARY
C IDL IS LINE SAMPLING ON DISK
C IDE IS ELEMENT SAMPLING ON DISK
C IS IS STARTING SECTOR FOR REQUEST
C IF IS = -1 IMP NON-HIT
C IM IS WORD OFFSET FROM STARTING SECTOR
      DIMENSION J(112)
      DATA N/0/
      IS=-1
      IF(N.EQ.0)CALL OPN(7)
      N=1
      CALL RFW(7)
      CALL READW(7,112,J)
      DO 1 K=1,8
      L=(K-1)*14+1
      IF(ID.NE.J(L))GO TO 1
      L=L+1
      IF(IE.NE.J(L))GOTO 1
      L=L+1
      ILS=J(L)
      L=L+1
      IES=J(L)
      L=L+1
      LD=J(L)
      L=L+1
      IED=J(L)
      IF(IL.LT.ILS)GOTO 1
      IF(IE.LT.IES)GOTO 1
      IF(IL.GT.ILS+LD*500)GOTO 1
      IF(IE.GT.IES+IED*224*3)GOTO 1
C      * * * (IL,IE) NOW LIES WITHIN AREA OF INTEREST * * *
      INC=3*IED
      M=(IL-ILS)/LD
      IL=ILS+M*LD
      IS=2*M
      IM=(IE-IES)/INC
      IE=IES+IM*INC
      IDL=LD
      IDE=IED
      IS=IS+1000*(K-1)
      CALL AGE(K)
      IF(IM.IT.112)GO TO 1
      IM=IM-112
      IS=IS+1
```

212

```

$JOB INIT
$REW,5
$INCLUDE IO
$INCLUDE FIO
$ASSIGN 6 0
$FORTRAN
    DIMENSION JSEC(112)
    COMMON/SYSCOM/NDAT(100)
101  FORMAT(' TYPE IN 4 DIGIT TIME, HOURS AND MINUTES')
102  FORMAT(2I2)
901  FORMAT('//1X,8A3//')
    CALL OPN(2)
    CALL SCRA(2,0)
    CALL READW(2,112,JSEC)
    CALL CLOS(2)
    CALL FROGS(1,5HOPCOM,7)
    WRITE(1,901)(JSEC(JJ),JJ=1,8)
    WRITE(1,101)
    READ(1,102)I1,I2
    DO 99 JJ=1,100
99   NDAT(JJ)=0
      I=60*I2+3600*I1
      NDAT(96)=I
    CALL EXIT
    ENDS
$CATALOG
TYPE=FG
NAME=INIT,2
$SSIGN 1=1,2=VERS,3=SYSCOM
BEGIN
$EOJ

```

```

$JOB LNDMRK
$ASSIGN,6,0
$REW,5
$INCLUDE TVSAT
$INCLUDE WAIT
$INCLUDE LOOKUP
$INCLUDE CSF
$INCLUDE FIO
$INCLUDE IO
$FORTRAN
    DIMENSION MIN(8),MOUT(10),NAM(2)
    DATA NAM/6HLNDRMK/
    DATA MOUT/6HDLNDRM/
C INPUT IS FRAME, LAT, LON, TVL, TVE, KEY
C OUTPUT IS SYDDDD, HHMMSS, KEY, SATL, SATE, LAT, LON
1   MIN(1)=NAM(1)
    MIN(2)=NAM(2)
    CALL IQ(MIN)
    IF(MIN(1).EQ.NAM(1))CALL EXIT
C
    CALL TVSAT(MIN(1),MIN(4),MIN(5),IL,IE,IT,ID)
C ID=0 IMP MISSING FRAME OR TAPE LINKAGE
    IF(ID.EQ.0)CALL EXIT
C SYDDDD
    MOUT(3)=ID
C HHMMSS
    MOUT(4)=IT
C KEY
    MOUT(5)=MIN(6)
C SATLIN
    MOUT(6)=IL
C SAT ELEMENT
    MOUT(7)=IE
C LAT
    MOUT(8)=MIN(2)
C LON
    MOUT(9)=MIN(3)
C KIND
    MOUT(10)=MIN(7)
    CALL SQ(MOUT)
    GOTO 1
END$
$CATALOG
TYPE=FG
NAME=LNDMRK
ASSIGN 2,REELS,3=FRAMES
BEGIN
$EOJ

```

214

```

$JOB LFRAME
$ASSIGN,6,0
$REW,5
$INCLUDE CSF
$INCLUDE FIO
$INCLUDE FTIME
$INCLUDE GETGAM
$INCLUDE ILALO
$INCLUDE IO
$INCLUDE LOOKUP
$INCLUDE SATFAR
$INCLUDE TVSAT
$INCLUDE WAIT
$FORTRAN
    DIMENSION MES(8),NAM(2)
    DATA NAM/6HLFRAME/
1    MES(1)=NAM(1)
    MES(2)=NAM(2)
    CALL IO(MES)
    IF(MES(1).EQ.NAM(1).AND.MES(2).EQ.NAM(2))CALL EXIT
    DO 2 I=1,8
    IF(MES(I).NE.0)CALL DOIT(MES(I))
2    CONTINUE
    GO TO 1
    END
    SUBROUTINE DOIT(N)
    COMMON/SYSCOM/ICOM(100)
    EQUIVALENCE (ICOM(2),IDAY)
    DIMENSION M(14),MOUT(24)
    CALL LOOK(3,N,M)
    ITL=M(5)
    ITE=M(6)
    CALL TVSAT(N,ITL,ITE,IL2,IF2,IT,ID)
    IF(ID.EQ.0)CALL EXIT
    NTAPE=M(2)
    NLIN=M(7)
    ENCODE(72,100,MOUT)N,NTAPE,ID,IT,IL2,IE2,NLIN
100  FORMAT('F',I5,2I7,18,3I7,2I9)
    IF(IDAY.NE.ID)GO TO 1
    INAV=1
    PICTIM=FTIME(IT)
    XLIN=IL2
    XELE=IF2
    CALL GETGAM(ID,IT,BETA IN,BETDOT)
    CALL SATFAR(PICTIM,XLIN,XELE,XLAT,XLON,1,INAV,BETA IN,BETDOT,0.0)
    ILAT=ILALO(XLAT)
    ILON=ILALO(XLON)
    ENCODE(72,100,MOUT)N,NTAPE,ID,IT,IL2,IE2,NLIN,ILAT,ILON
1    CALL TO(MOUT)
    RETURN
    ENDS
$CATALOG
TYPE=FG
NAME=LFRAME
ASSIGN 2=REELS,3=FRAMES,11=LANDMA
BEGIN
$EOJ

```

```

$JOB LISDIR
$ASSIGN 6 0
$REW,5
$INCLUDE CSF
$INCLUDE IO
$INCLUDE FIO
$FORTRA
    DIMENSION MES(8),J(112)
    DIMENSION MOUT(24)
    COMMON/SYSCOM/NDAT(100)
    DATA MFS/6HLISDIR/
    CALL IO(MES)
    CALL OPN(2)
    CALL RFW(2)
    CALL READW(2,112,J)
    CALL CLOS(2)
1  FORMAT(13,517,313;15)
    DO 11 K=1,8
    L=14*(K-1)+1
    M=L+6
    MK=NDAT(K+82)
    MJ=J(L+7)
    IF(J(L).EQ.0)GOTO 11
    ENCODE(72,1,MOUT)K,MJ,(J(N),N=L,M),MK
    CALL TO(MOUT)
11 CONTINUE
    CALL EXIT
    ENDS
$CATALOG
TYPE=FG
A =
VSWIGPISDIR
BEGIN
$EOJ

```

```
203  CONTINUE
      CALL LISTEM(JDIR,MES(1),12)
      GOTO 199
C BETA-BETADOT
204  CONTINUE
      CALL LISTEM(JDIR,MES(1),14)
      GOTO 199
C EARTH EDGES
205  CONTINUE
      CALL LISTEM(JDIR,MES(1),15)
      GOTO 199
      ENDS
$CATALOG
TYPE=FG
NAME=LISLMK
ASSIGN 2=LANDMA
BEGIN
$EOJ
```

```

      IF(KIND.NE.12)GOTO 45
      IH=IH+1
      I1=JDIR(JA+2)
      I2=JDIR(JA+3)
      I3=JDIR(JA+4)
      I4=JDIR(JA+5)
      GOTO 44
45    IF(KIND.NE.13)GOTO 44
      IH=IH+1
      I5=JDIR(JA+2)
      I6=JDIR(JA+3)
      I7=JDIR(JA+4)
44    CONTINUE
      ENCODE(72,339,MOUT)I1,I2,I3,I4,I5,I6,I7
      IF(IH.EQ.2)CALL TQ(MOUT)
339   FORMAT('0',8I8)
      RETURN
      END
      DIMENSION JDIR(560),MES(8),MOUT(24),MER(24)
      DIMENSION NAM(2)
      COMMON/SYSCOM/NDAT(100)
      DATA NAM/6HLISLMKZ
      DATA MER/30H NO LANDMARKS ON SPECIFIED DAY
199   MES(1)=NAM(1)
      MES(2)=NAM(2)
      CALL IQ(MES)
      IF(MES(1).EQ.NAM(1))GO TO 299
      IF(MES(1).EQ.0)CALL ABORT
      IF(NDAT(92).LT.0)CALL WAIT(NDAT(92))
      NDAT(92)=-1
      CALL OPN(2)
      MSEC=5*MOD(MES(1),100)
      CALL SCRA(2,MSEC)
      CALL READW(2,560,JDIR)
      CALL CLOS(2)
      NDAT(92)=0
      JUMP=MES(2)
      IF(JUMP.NE.0)GOTO 4444
      DO 450 JA=1,4
      JB=13-JA
450   CALL LISTEM(JDIR,MES(1),JB)
      GOTO 199
4444  IF(JUMP.LT.1)CALL EXIT
      IF(JUMP.GT.5)CALL EXIT
      GOTO(203,202,201,204,205),JUMP
C LIST LANDMARKS
201   CONTINUE
      CALL LISTEM(JDIR,MES(1),9)
      GO TO 199
299   CONTINUE
      CALL EXIT
C LIST SPIN RATE
202   CONTINUE
      CALL LISTEM(JDIR,MES(1),11)
      GO TO 199
C LIST ORBIT

```

218


```
KIND=JDIR(JA+6)/4096
```

219

```

$JOB EDAY
$ASSIGN,6,0
$REW,5
$INCLUDE FIO
$INCLUDE CSF
$INCLUDE WAIT
$INCLUDE IO
$FORTRAN
    DIMENSION MES(8),J(560),MER(24)
    COMMON/SYSCOM/NDAT(100)
    DATA MES/6HEDAY /
    DATA MER/30HPRIVILEGED FUNCTION REFUSED
    CALL IO(MES)
    IF(NDAT(95).LT.0)GOTO 100
    CALL TO(MER)
    CALL EXIT
100  IF(NDAT(92).LT.0)CALL WAIT(NDAT(92))
    NDAT(92)=-1
    CALL OPN(2)
    N=MES(1)
    NSEC=5*MOD(N,100)
    CALL SCRA(2,NSEC)
    CALL READW(2,560,J)
    DO 200 L=1,560,7
    IF(J(L).EQ.N)J(L)=0
200  CONTINUE
    CALL SCRA(2,NSEC)
    CALL WRITW(2,560,J)
    CALL CLOS(2)
    NDAT(92)=0
    CALL EXIT
    ENDS
$CATALOG
TYPE=FG
NAME=EDAY
ASSIGN 2=LANDMA
BEGIN
DEOJ

```

```

      INAV=1
      DO 1 J=1,500
      IF(ICOM(1).EQ.0)GOTO 222
C READ TAPE RECORD
      CALL READW(2,2731,IT)
C MOVE SAMPLES SPLIT TO IS
C FIRST TAPE WORD TO START SPLITTING
      ITST=1+(IES-1)/3
      NSPLIT=NF+6
C OFFSET BETWEEN SPLIT START AND FIRST WANTED ELEMENT
      IR=MOD(IFS+2,3)
C PLACE IN SATELLITE COORDINATE SYSTEM TO PUT SPLIT SAMPLES
      IDEST=IOF+IES-IR -1
      IF(KEE.EQ.0)CALL CRACK(NSPLIT,IT(ITST),IS(IDEST))
C IF EEC NEEDED, MUST SPLIT WHOLE LINE
      IF(KEE.NF.0)CALL CRACK(8193-IOF,IT,IS(IOF))
C DO EARTH-EDGE CORRECTION
      CALL EDGFCO(PST,KEE,INAV,BETA,BDOT,IPL,IS,IDEL,IBAD)
      IF(IBAD.EQ.0)GOTO 777
      DO 776 J7=1,8193
776  IS(J7)=0
777  CONTINUE
      IPL=IPI+ILD
C K IS PLACE IN OUTPUT BUFFER TO PUT LINE
      K=MOD(J,5)
      IF(K.EQ.0)K=5
      K=K-1
      KK=0
      JOF=IOF+IDELT
      DO 2 JJ=IES,IEL,IED
      M=0
      JA=JOF-1+JJ
      DO 22 JJJ=1,IED
      MM=JA+JJJ-1
22  M=M+IS(MM)
      M=M/IED
      KK=KK+1
2  IS(KK)=M
      KAD=224*K+1
      CALL PACK(672,IS,ID(KAD))
      IF(K.NF.4)GO TO 11
      CALL SCRA(3,NSEC)
      CALL WRITW(3,1120,ID)
      NSEC=NSEC+10
11  IF(ILD.EQ.1)GOTO 1
      II=ILD-1
      DO 12 IX=1,II
12  CALL READW(2,2,IT)
1  CONTINUE
      CALL TQ(MOUT)
222 CALL RFW(2)
      CALL CLOS(2)
      CALL CLOS(3)
      CALL EXIT
      ENDS

```

221

\$CATALOG
NAME=ABSLD
TYPE=FG
TBSIZE=5000
ASSIGN 2=11,3=DK,4=REELS,11=LANDMA
BEGIN
\$EOJ

\$JOB LOGGIN
\$ASSIGN 6 0
\$REW,5
\$FORTRAN

```
COMMON/SYSCOM/N(100)  
DIMENSION MESX9(10)  
DIMENSION MES(8)  
DIMENSION MME1(2),MME2(10)  
DIMENSION MES2(24),MFS3(24),MES4(24),MES5(24),MOUT(24)  
DIMENSION MC(100)  
DIMENSION MPROJ(112)  
DIMENSION MMOUT(13)  
EQUIVALENCE(MOUT(11),MMOUT(1))  
DATA MC/100*0/  
DATA MFSX9/6HDOIOXX/  
DATA MFS5/30H***PRIVILEGED PROJECT***  
DATA IRLK/3H /  
DATA MFS/6HLOGGIN/  
DATA MFS4/30HILLEGAL PROJECT  
DATA MFS2/30HMUST LOGOUT FIRST  
DATA MFS3/6HLOGOUT/  
DATA MME1/6HENH001/  
DATA MME2/0,0,0,63,0,63,0,63,0,63/  
MES3(3)=0
```

```
MME2(1)=MME1(1)  
MME2(2)=MME1(2)  
CALL OPN(4)  
CALL RFW(4)  
CALL RFADW(4,90,MC)  
CALL CLOS(4)
```

C COMMON 1-90 FROM DISK
C COMMON 91-94 ZERO
C COMMON 95-100 UNCHANGED

DO 983 JA=1,95

```
983 N(JA)=MC(JA)  
CALL IQ(MES)  
IF(MES(1).EQ.0.OR.MES(1).GT.9999)CALL EXIT  
IF(N(1).EQ.0)GO TO 111  
CALL TQ(MES2)
```

```
CALL EXIT  
111 CONTINUE  
N(80)=0  
N(79)=177  
N(78)=1400  
N(76)=2  
N(75)=100410000  
N(74)=0  
N(73)=0  
N(72)=102020202
```

```
CALL SQ(MESX9)  
CALL SQ(MME2)  
NTICK=MES(2)  
IF(NTICK.EQ.0)NTICK=10  
NTICK=NTICK*60  
NTICK=NTICK*120  
CALL FROGS(6,6HLOGOUT,254,NTICK
```

)

223

```

CALL OPN(3)
CALL RFW(3)
CALL READW(3,10,MOUT)
CALL CLOS(3)
N(1)=MFS(1)
N(96)=N(96)+N(97)
N(97)=0
IF(N(96).GT.86400)N(96)=N(96)-86400
MM=N(96)/60
MM=MOD(MM,60)
M1=MOD(MM,10)
M2=MOD(MM/10,10)
MM=N(96)/3600
M3=MOD(MM,10)
M4=MM/10
ENCODE(42,222,MMOUT,M4,M3,M2,M1
222 FORMAT(' ACTIVE AT ',I2,3I1)
CALL TO(MOUT)
CALL OPN(2)
CALL SCRA(2,0)
CALL READW(2,112,M PROJ)
CALL CLOS(2)
DO 93 JJ=1,112,2
IF(M PROJ(JJ).EQ.N(11))GO TO 993
93 CONTINUE
CALL TO(MES4)
CALL SQ(MES3)
CALL EXIT
993 N(95)=M PROJ(JJ+1)
IKEY=N(95).AND.'40000000
IF(IKEY.FQ.0)CALL EXIT
CALL TO(MES5)
CALL EXIT
END$
$INCLUDE IO
$INCLUDE FIO
$INCLUDE CSF
$CATALOG
TYPE=FG
NAME=LOGGIN
ASSIGN 2=PROJS,3=VERS,4=SYSCOM
BEGIN
$EOJ

```

224

```

$JOB DOIOXX
$REW,5
$ASSIGN,6,0
$INCLUDE CSF
$INCLUDE HIO
$FORTRA
    DIMENSION MES(8)
    COMMON/SYSCOM/N(100)
    DATA MFS/6HDOIOXX/
    CALL IQ(MES)
    CALL HIO('101','2000+16,N(80))
    CALL HIO('101','2000+2*16,N(79))
    CALL HIO('101','2000+3*16,N(78))
    CALL HIO('101','2000+5*16,N(76))
    CALL HIO('101','2000+6*16,N(75))
    CALL HIO('101','1000+3*16,N(74))
    CALL HIO('101','1000+5*16,N(73))
    CALL HIO('101','1000+6*16,N(72))
    CALL EXIT
END$
$CATALOG
TYPE=FG
NAME=DOIOXX
BEGIN
$EOJ

```

\$JOB CNTRL
\$REW,5
\$ASSIGN,6,0
\$ASSEMB

START BLL \$MAIN
BLU \$EXIT
END\$ START

\$FORTRAN

SUBROUTINE MAIN
DIMENSION MES(8),NAM(2),MOUT(24)
DIMENSION MSYS(10)
DIMENSION LMES(3)
DIMENSION LF(2)
DIMENSION NF(2)
DIMENSION IW(2)
DIMENSION LD(2)
DIMENSION JE(2)
DIMENSION LKMES(3)
DIMENSION MTVL(10)
DIMENSION MECHO(24)

C NDATA(55) IS CURSOR POINTER WORD
C NDATA(56) IS CURRENT-FRAME/DESIRED FRAME
C 57 IS FRAME BOUNDS
C 59 IS LAT
C 60 IS LONG
C 61 IS WIND TEMP STORAGE
C 62 IS LAST AREA USED FOR A LOAD
C 63

C 64 IS CURSOR SIZE
COMMON/SYSCOM/NDATA(100)
DATA MECHO/3H * /
DATA IW/6HBAKGND/
DATA NF/6HNAVFRM/
DATA LF/6HLFRAME/
DATA LD/6HLNDRMK/
DATA JF/6HFRMEAR/
DATA LMES/6HLOOP /
DATA LKMES/6HLISDIR/
DATA MTVL/6HLCNTV/
DATA MOUT/30H BAD CNTRL
DATA NAM/6HCNTRL /

999 MES(1)=NAM(1)
MES(2)=NAM(2)
CALL IQ(MES)
K=MES(1)
IF(MES(1).EQ.NAM(1))RETURN
IF(NDATA(1).EQ.0)GOTO 999
MECHO(1)=MECHO(1).AND.'77777400
MECHO(1)=MECHO(1).OR.K

C L
IF(K.NF.'314)GOTO 2
NDATA(62)=0
CALL SQ(LMES)

226


```

      GOTO 1
C   A
  2   IF(K.NF.'301')GOTO 3
      NDAT(62)=0
      CALL STEP
      GOTO 1
C   B
  3   IF(K.NF.'302')GOTO 4
      NDAT(62)=0
      CALL BACK
      GOTO 1
C   R
  4   IF(K.NF.'322')GOTO 5
      NDAT(62)=0
      CALL RFST
      GOTO 1
C   C
  5   IF(K.NF.'303')GOTO 6
      MSYS(1)=IF(1)
      MSYS(2)=IF(2)
      MSYS(3)=NDAT(56).AND.'7777
      DO 500 JA=4,10
500  MSYS(JA)=0
      CALL SQ(MSYS)
      GOTO 1
C   M
  6   IF(K.NF.'315')GOTO 7
      NDAT(62)=0
      MSYS(1)=LD(1)
      MSYS(2)=LD(2)
      MSYS(3)=NDAT(56).AND.'7777
      MSYS(4)=NDAT(59)
      MSYS(5)=NDAT(60)
      MSYS(6)=NDAT(55)/4096
      MSYS(7)=MOD(NDAT(55),4096)
      MSYS(8)=0
      CALL SQ(MSYS)
      CALL STEP
      GOTO 1
C   E
  7   IF(K.NF.'305')GOTO 8
      MSYS(1)=JE(1)
      MSYS(2)=JE(2)
      MSYS(3)=NDAT(56).AND.'7777
      MSYS(4)=NDAT(55)/4096
      MSYS(5)=MOD(NDAT(55),4096)
      CALL SQ(MSYS)
      GOTO 1
C   N
  8   IF(K.NF.'316')GOTO 9
      CALL SQ(NF)
      GOTO 1
C   W
  9   IF(K.NF.'327')GOTO 10
      NDAT(62)=0
      IF(NDAT(61).NE.0)GOTO 900
      NDAT(61)=NDAT(55)

```

```

CALL STEP
GOTO 1
900 MSYS(1)=IW(1)
MSYS(2)=IW(2)
MSYS(6)=NDAT(56).AND.'7777
MSYS(3)=MSYS(6)-1
IT=NDAT(57)/4096
IF(MSYS(6).EQ.IT)CALL STEP
KK=NDAT(61)
NDAT(61)=0
MSYS(4)=KK/4096
MSYS(5)=MOD(KK,4096)
MSYS(7)=NDAT(55)/4096
MSYS(8)=MOD(NDAT(55),4096)
MSYS(9)=NDAT(64)/4096
MSYS(10)=MOD(NDAT(64),4096)
CALL SQ(MSYS)
GOTO 1
C B
10 IF(K.NF.'304)GOTO 11
CALL SQ(LKMES)
GOTO 1
C 1 THRU 8
11 IF(K.LT.'260)GOTO 12
IF(K.GT.'270)GOTO 12
MTVL(3)=K-'260
NDAT(62)=MTVL(3)
MTVL(4)=NDAT(56).AND.'7777
MTVL(5)=0
MTVL(6)=0
MTVL(7)=1
CALL SQ(MTVL)
GOTO 1
C J K
12 IF(K.LT.'311.OR.K.GT.'313)GOTO 13
K=K-'307
MTVL(3)=NDAT(62)
MTVL(4)=NDAT(56).AND.'7777
MTVL(7)=K
NDAT(62)=0
C CURSOR
ICL=NDAT(55)/4096
ICE=MOD(NDAT(55),4096)
ISZ=500/K
ISZ=ISZ/2
MTVL(5)=ICL-ISZ
ISZ=700/K
ISZ=ISZ/2
MTVL(6)=ICE-ISZ
CALL SQ(MTVL)
GOTO 1
13 CONTINUE
GOTO 999
1 CALL TO(MECHO)
GOTO 999

```

228

```

END
SUBROUTINE STEP
COMMON/SYSCOM/NDAT(100)
IT=NDAT(57)/4096
IB=NDAT(57)-4096*IT
ID=MOD(NDAT(56),4096)
L=IT-IR
IF(ID.EQ.IT)GOTO 2
NDAT(56)=NDAT(56)+1
RETURN
2 NDAT(56)=NDAT(56)-L
RETURN
END
SUBROUTINE BACK
COMMON/SYSCOM/NDAT(100)
IT=NDAT(57)/4096
IB=NDAT(57).AND.'7777'
ID=NDAT(56).AND.'7777'
L=IT-IR
IF(ID.EQ.IB)GOTO 2
NDAT(56)=NDAT(56)-1
RETURN
2 NDAT(56)=NDAT(56)+L
RETURN
END$
$INCLUDE CSF
$CATALOG
TYPE=FG
NAME=CNTRL
BEGIN
$EOJ

```

```

$JOB CC
$REW,5
$ASSIGN 6,0
$ASSEMB
START   BLL    $MAIN
        BLU    $EXI
        ENDS   START
$INCLUDE,CSF
$INCLUDE,STAGE
$INCLUDE,IO
$FORTRAN
    SUBROUTINE MAIN
    DIMENSION LIS(300)
    DIMENSION MES(27),M(60)
    DIMENSION NERN(2)
    DIMENSION NER(2)
    DIMENSION M3(24)
    DIMENSION NAM(2)
    COMMON/SYSCOM/IP(100)
    DATA M3/6HCD FIN/
    DATA NER/6HERROR /
    DATA NERN/6HCRAP: /
    DATA ISTAR/3H*** /
    DATA IRLK/'40/
    DATA IFN/3HEND/
    DATA NAM/6HCC /
    MES(1)=NAM(1)
    MES(2)=NAM(2)
    CALL IQ(MES)
    CALL CRACK(1,ISTAR,ISTAR)
    CALL IO1('0707)
    CALL IO1('0207)
    CALL IO1('0216)
    CALL IO('0203,300,LIS)
    CALL IO1('0210)
111  CALL IO('0701,27,MES)
    CALL IO1('0700)
    IF(IP(1).EQ.0)GOTO 111
    CALL IO1('0607)
    CALL IO('0602,27,MES)
    CALL IO1('0600)
    CALL IO1('0610)
    IF(IEN.EQ.MES(1))GO TO 1000
    CALL CRACK(60,MES,M)
    KO=0
    DO 20 J=1,300,3
    IF(MES(1).NE.LIS(J))GO TO 20
    KO=J
    M1=LIS(J+1)
    M2=LIS(J+2)
20  CONTINUE
    IF(KO)302,302,301
301  CONTINUE
    IF(M(3).FQ.ISTAR)GO TO 100
    IF(M(3).FQ.IBLK)GO TO 200

```

230

```

300  CONTINUE
    MES(1)=NFR(1)
    MES(2)=NFR(2)
303  CONTINUE
    CALL SQUASH(M,MES(3))
    CALL TO(MES)
    GO TO 111
302  MES(1)=NFRN(1)
    MES(2)=NFRN(2)
    GO TO 303
100  MES(1)=M1
    MES(2)=M2
    CALL SQUASH(M(2),MES(3))
    CALL SQ(MES)
    GO TO 111
200  MES(1)=M1
    MES(2)=M2
    I=2
    DO 201 J=3,10
    CALL MAKE(I,M(3),N,L)
    IF(L.NF.0)GO TO 300
201  MES(J)=N
    CALL SQ(MES)
    GO TO 111
1000 CALL IO1('0710)
    CALL TO(M3)
    RETURN
    END
    SUBROUTINE SQUASH(IS,ID)
    DIMENSION IS(24),ID(8)
    CALL PACK(24,IS,ID)
    RETURN
    END
    SUBROUTINE MAKE(I,M,N,L)
    DIMENSION M(24)
    DATA ICOM/'40/
    DATA MIN/'55/
    N=0
    L=0
    ISGN=1
1   IF(I.GT.60)GO TO 100
    IF(M(I).EQ.MIN)GO TO 900
    IF(M(I).EQ.ICOM)GO TO 100
    KK=M(I)-48
    IF(KK.LT.0.OR.KK.GT.9)L=1
    N=10*N+KK
    I=I+1
    GO TO 1
100  I=I+1
    N=N*ISGN
101  RETURN
900  I=I+1
    ISGN=-1
    IF(N.NF.0)L=1
    GO TO 1
    ENDS
$ASSIGN,6,17

```

231

\$CATALOG
TYPE=FG
NAME=CC
ASSIGN 2=NAMLIS,6=1,7=7
BEGIN
\$EOJ

232

```

$JOB NAMLIS
$ASSIGN 6 0
$REW,5
$INCLUDE IO
$INCLUDE FIO
$
  FORTRE-MENSION J(300),M(3)
    DATA IFN/3HEND/
    CALL OPN(7)
    L=0
  1   CALL RDANW(7,3,M)
    IF(M(1).EQ.IEN)GOTO100
    L=L+1
    J(L)=M(1)
    L=L+1
    J(L)=M(2)
    L=L+1
    J(L)=M(3)
    GO TO 1
  100 DO 200 LL=L,299
  200 J(LL+1)=0
    CALL OPN(2)
    CALL REW(2)
    CALL WRANW(2,300,J)
    CALL CLOS(2)
    CALL EXIT
  ENDS

```

```

$ASSIGN 2 NAMLIS
$CATGO
BJ LOGGIN
EJ LOGOUT
AT ACOTAP
BS LAGSIZ
CR CC
BB DLIM
DE DEFPNT
BF DFRAME
BK LDCNTR
BL DLANDM
DO DORBIT
BR DRATE
DS DSRATE
DT DEFTAP
DV DVERS
EK ERADIR
ED EDAY
EL DELMRK
EX EXITWL
E1 ENH001
GL LNDMRK
IF LFRAME
IT LISTAP
KS SETKEY
LF FINFRM

```

233

LK LISDIR
LD LISLMK
LT FINDTP
MC USECUT
MF USEFIL
MG USEGRA
ML USELAP
MR USERAD
MS USESPA
NA NAVSYS
NC NAVCHG
NT XFORMS
PL SURPLT
QM MINBND
QP PEAKCO
QS SURFCO
RD ERTSLD
RF RESFRM
RK RESTDK
SF SETF
SK SAVEDK
SS STATUS
TB DEFMET
TD NRZDEC
TE FRMEAR
TS EARSAT
TT EARTAP
WC WCELL
WD FILDIM
WE FILELE
WI BAKGND
WL FILLIN
WP FILWGT
WT WHATAP
XF LDCNTV
ZA*ANNOT
ZD ANNDIS
ZE ANNINT
ZM*PLTMAP
ZZ LOGDUM
END
\$EOJ

234

Page intentionally left blank

```

C GET AREA DIRECTORY ENTRY
  CALL OPN(5)
  CALL RFW(5)
  CALL READW(5,112,J)
  IP=14*(IA-1)
  DO 1 JA=1,14
    IP=IP+1
1   MD(JA)=J(IP)
    CALL CLOS(5)
    IF(MD(1).LE.0)GOTO 100
C PERFORM TAPE LOOKUP
  CALL LOOK(2,MD(8),MF)
  IF(MT(1).EQ.0)GOTO 100
C LOOK UP FRAME ENTRY FOR PROJECT PROTECT KEY ONLY
  CALL LOOK(3,IF,MF)
  IF(MF(14).EQ.0)GOTO 2
  IF(MF(14).EQ.NDAT(1))GOTO 2
  IF(NDAT(95).LT.0)GOTO 2
  CALL SQ(MESO)
  GOTO 100
C TAPE
2   CONTINUE
  MESO(4)=MT(1)
C STARTING LINE OF TAPE
  ILF=MT(4)
C STARTING LINE OF AREA
  ILF=MD(3)-ILF+1
C OFFSET
  ILF=ILF+MD(5)*IL
  MESO(5)=ILF
C SAME WITH ELEMENTS
  IEF=MD(4)-MT(5)+1+MB(6)*IE
  MESO(6)=IEF
C NUMBER OF LINES
  MESO(9)=500/NR
C TV RELATIVE
  MESO(7)=ITLS
  MESO(8)=ITES
C DO ACTUAL LOAD
  MSEC=1000*(IA-1)+2*IL
C NSEC IS POINTER TO 5 LINE AREA ON DIGITAL DISK
  NSEC=MSEC
  CALL TVI(1)
  DO 96 JA=1,700
96   L(JA)=0
      JA=ITLS-1
      IF(ITLS.EQ.1)GOTO 97
      DO 95 KL=1,JA
95   CALL TVD(L)
C KL IS SCREEN LINE POINTER
97   KL=ITLS
      CALL OPN(4)
10  CALL SCRA(4,NSEC)
      CALL READW(4,1120,K)
      DO 11 JA=1,5
      JB=224*(JA-1)+1

```

236

```

      CALL CRACK(672,K(JB),J)
C JC IS SOURCE ELEMENT POINTER
      JC=IE
C KT IS SCREEN ELEMENT POINTER
      KT=ITES-1
14   DO 13 JD=1,NR
      KT=KT+1
      IF(KT.GT.700)GOTO 15
      JZ=J(JC)+1
13   L(KT)=ITAB(JZ)
      JC=JC+1
      IF(JC.GT.672)GOTO 15
      GOTO 14
15   DO 12 JB=1,NR
      CALL TVD(L)
      KL=KL+1
      IF(KL.EQ.500)GOTO 90
12   CONTINUE
11   CONTINUE
      NSEC=NSEC+10
      IF(NSEC.GT.MSEC+999)GOTO 90
      GOTO 10
90   DO 91 JA=4,10
91   MESJ(JA)=0
      CALL TVE
      CALL SQ(MESO)
      CALL SQ(MESJ)
      CALL EXIT
100  CALL TQ(MESE)
      CALL EXIT
      ENDS
$CATALOG
TYPE=FG
NAME=LDCNTV
ASSIGN 2=REELS,3=FRAMES,4=DK,5=DKDIR
BEGIN
$EOJ
$EOJ
$FIN

NEXT JOB

```